Prode Properties

Properties of pure fluids and mixtures

User's Manual rel. 1.28

Copyright Prode Milano Italy

www.prode.com

TABLE OF CONTENTS

License Agreement	4
How to obtain technical support	4
How to contact Prode	4
Introduction to Prode Properties	5
Install the software	6
Require a software license	7
Activate a software license	7
Prode Properties : initial setup	8
Getting Started from Microsoft Excel (Windows version)	9
Getting Started from Microsoft Excel (predefined examples)	10
Getting Started from Microsoft Excel (working with macros)	13
Getting started from LibreOffice	14
Getting started from Python	15
Getting started from Python (continuation)	16
Getting started from MATLAB	17
Accessing Prode Properties library (with programming languages)	19
Prode Properties : introduction	20
Prode Editor : Introduction	21
Prode Editor : Feeds page	22
Prode Editor : Streams page	24
Prode Editor : Config page	25
Prode Editor : Chemicals page	27
Prode Editor : BIPs page	29
Prode Editor : Models page	31
Prode Editor examples : define / edit streams	32
Prode Editor examples : set units of measurement and configuration settings	34
Prode Editor examples : inspect streams, solve flash operations	35
Prode Editor examples : edit BIPs, calculate new BIPs	37
Working with archives	42
Introducing Prode Properties library methods	43
Methods for thermodynamic calc's	43
Methods for stream's data access	45
Methods to work with packages	50
Methods to work with streams	51
Methods to define stream's operating conditions	53
Methods for solving staged columns	54
Methods for solving reactors	56
Methods for solving fluid flow problems	56
Methods for Hydrates phase equilibria	56
Methods for solving Polytropic operations	57
Methods to design / rate orifices and relief valves	57
Methods for calculating equilibrium lines in phase diagrams	58
Methods for direct access to properties (F.H.S.V) and derivatives (T.P.W)	59
Methods for stream's data access	60
Methods for chemical's file access	63
Methods to set / access options / settings	66
Table of codes to specify the different models	67
Auxiliary methods	68
Methods to access Model's data	60
Methods to control error's messages	60
Methods for accessing Prode Editor	60
Methods to load / save archives	70
Methods for accessing / defining the units of measurement	70

72
73
74
75
78
79
80

License Agreement

Agreement made between Prode "Prode" and "User".

• Prode is the owner of the product "Prode Properties" including , but not limited to, dynamic link libraries, static libraries, header files, sample programs, utility programs, together with the accompanying documentation collectively known as the "software",

· User desires to obtain the right to utilize the software, the parties hereby agree as follows

Personal license

A version with limited features is available for personal use at home or in educational establishments for teaching purposes, all other applications, without first obtaining a commercial license from Prode, are expressly prohibited. Commercial license

Upon full payment of the license fee the User has full right to utilize the purchased number of units of the software, a unit is defined as one copy of the software or any portion thereof installed on one stand-alone computer, for networked computers one unit shall be applied for each user having concurrent access and one unit shall be applied for the server. For all applications

• Prode grants the nonexclusive, nontransferable right to use the software.

• User has a royalty free right to reproduce and distribute the software as available from Prode Internet server provided that User doesn't remove or alter any part of the software or of the licensing codes and threat the software as a whole unit.

• You cannot decompile, disassemble or reverse engineer the files containing the licensed software, or any backup copy, in whole or in part.

• You cannot rent, lease or sublicense the Licensed Software without express agreement by Prode.

• The software is provided "as is, where is", Prode does not warrant that software is free from defects, or that any technical or support services provided by Prode will correct any defects which might exist.

• Prode shall not be liable for any damages that may result directly or indirectly from the use of these software programs including any loss of profits, loss of revenues, loss of data, or any incidental or consequential damages that may arise out of use of these software.

• Your license is effective upon your acceptance of this agreement and installing the Licensed Software.

• This license agreement shall remain in effect until the Licensed Software will be in use.

• You may terminate it at any time by destroying the Licensed Software together with all copies. It will also terminate if you fail to comply with any term or condition of this Agreement. You agree upon such termination to destroy all copies of the Licensed Software in any form in your possession or under your control.

• Prode will provide the licensee with limited technical support by telephone, or by electronic media for a period of 60 days after delivery of the product.

Limits of free version (no license installed)

- 1) Limited database of components (with license > 1650 chemicals)
- 2) Max 10 components per stream (with license > 100 components per stream)
- 3) The software stops after running for 1 hour (you can restart the app to recover)

How to obtain technical support

we welcome your comments or suggestions about our products, while the program has been tested carefully to ensure proper operation, it still may be possible for an unusual situation to result in an error. We will have a much greater chance of fixing or assisting with errors and problems if they are provided to us in a form that is repeatable. In reporting a problem to us, the following information should be given:

- customer reference
- the version of the software
- · a copy of the procedure you are running and if possible the input data
- a detailed description of what you were doing (sequence of operations) when the problem occurred
- any additional information you think may describe the problem

How to contact Prode

you can contact Prode by phone, web page or email, the details are available at http://www.prode.com

Introduction to Prode Properties

Prode Properties includes a comprehensive collection of procedures to solve problems in areas such as :

- Process Simulation, Process Control,
- Physical Properties Data, Data Analysis
- Equipment Design, Separations...

Technical features overview

- Prode Properties is a thermodynamic library written in C++ (ISO 2017) and released as compiled library
- Multiple threads are supported by design, no limits on number of concurrent threads
- Standard versions for Window, Linux and Android, several other versions including Web Assembly available on request

• Prode Properties can work with many applications such as Microsoft Excel, Open Office, Python, MATLAB, MathCad etc.

- Allows up to 100 streams with up to 100 components per stream (user can redefine)
- Several compilations of chemical data and BIPs available (user can edit / add new components and BIPs)
- · Comprehensive set of thermodynamic models (> 30 models available)
- Complete set of multiphase (VLSH) flash operations T-P, H-P, H-T, S-P, S-T, V-P, V-T, H-V, S-V, PF-P, PF-T, CF-P...
- Functions for calculating specific properties of mixtures (critical point, Cricodentherm, Cricondenbar, cloud point etc.)
- Functions for calculating fugacity, enthalpy, entropy, volume plus derivatives vs. temperature, pressure, composition
- Functions for solving operating blocks as valves, compressors, columns, mixers, pipes...

• Functions for calculating stream properties as density, conductivity, viscosity (gaseous and liquid phases) surface tension, speed of sound, Joule Thomson etc.

Reference Literature

Although Prode Properties may appear easy to utilize, a basic knowledge of applied thermodynamics is required for selecting the proper methods and critically evaluate the results, to support your work we suggest these books :

- Introduction to Chemical Engineering Thermodynamics by Smith, Van Ness, Abbott
- The Properties of Gases & Liquids, by Reid, Prausnitz, Poling
- Thermodynamic models for industrial applications by Kontogeorgis, Folas

History

version 1.01 (first commercial release of Prode Properties) distributed in 1992

Actual release

version 1.28, released on Dec. 2022

Roadmap

version 1.29 (new version) Q2 2025

Install the software

Prode Properties is available from this page

http://www.prode.com/en/download.htm

download the software, run the installer and follow the instruction ask Prode for a different versions (there are versions for Windows, Linux, Android, IOS)

Test the software

Prode Properties includes a desktop app (Prode Properties Tests), run the app and follow the instruction Enter 1 to start a series of tests (the user can select single or multiple threads and compare results) Enter 2 to open Prode Editor

👪 C:\Program	Files\Prode\props.exe
--------------	-----------------------

[pp] C.(Program Pries(Props.exe	U	\sim
Rel 1.28.41 License 4000 support ends 20241212 user PRODE		
based on results from std::thread::hardware_concurrency() module		
your machine could execute up to 16 threads simultaneously,		_
however, since CPUs can support multiple logical threads per physical core		
we recommend to do not exceed 8 concurrent threads		
Enter 0 : Exit ; 1 : Test Prode Properties ; 2 = Test Prode Editor ->1		
Enter Number of (Concurrent) Threads (1-50) -> 1		
Thread=1 Cycle=1 Test analytic derivatives dFg, dH, dS, dV / dT, dP		
Thread=1 Cycle=1 Test Isothermal Flash data set 1		
Thread=1 Cycle=1 Test Isothermal Flash data set 2		
Thread=1 Cycle=1 Test Isothermal Flash data set 3		
Thread=1 Cycle=1 Test vapor-liquid, vapor-liquid-liquid phase diagrams		
Thread=1 Cycle=1 Test phase equilibria bubble / dew points data set 1		
Thread=1 Cycle=1 Test Hydrate phase equilibria including inhibitors		
Thread=1 Cycle=1 Test data regression procedures		
Thread=1 Cycle=1 Test fluid properties (against experimental values) data set 1		
Thread=1 Cycle=1 Test fluid properties (against experimental values) data set 2		
Thread=1 Cycle=1 Test fluid properties (against experimental values) data set 3		
Thread=1 Cycle=1 Test fluid properties (against experimental values) data set 4		
Thread=1 Cycle=1 Test fluid properties (against experimental values) data set 5		
Thread=1 Cycle=1 Test fluid properties (against experimental values) data set 6		
Thread=1 Cycle=1 Test fluid properties (against experimental values) data set 7		
Thread=1 Cycle=1 Test fluid properties (against experimental values) data set 8		
Thread=1 Cycle=1 Test Unit conversions		
Thread=1 Cycle=1 Test distillation (vapor-liquid, lquid-liquid)		
Thread=1 Cycle=1 Test Nozzle		
Thread=1 Cycle=1 Test Mixer		
Thread=1 Cycle=1 Test Flash operations (H)		
Thread=1 Cycle=1 Test Flash operations (S)		
Thread=1 Cycle=1 Test Flash operations (V)		
Thread=1 Cycle=1 Test Flash operations (Mixed)		
Thread=1 Cycle=1 Test water and steam properties, data set 1		
Thread=1 Cycle=1 Test ater and steam properties, data set 2		
Thread=1 Cycle=1 Test vapor-liquid-liquid separator		

With option 1) the procedure executes a series of automatic tests (solving hundreds of predefined problems) then it reports errors and problems.

The user can start multiple threads and compare results, note that Prode Properties supports any number of concurrent threads,

the table shows the number of seconds required to complete the tests on a Windows 11-64 computer with CPU AMD 4750U

Nr. threads	Time (seco	onds) relative speed	(Time single thread	* number of threads) /	Time multi-threaded cycle
1	8	1			
2	9	1.77			
4	10	3.2			
8	18	3.55			
16	32	4			
32	62	4.12			

the CPU AMD 4750U has 8 physical cores, the results show that relative speed do not increase much for any number of threads >8

Running the same tests on a CPU with 16 physical cores the application returns a relative speed of about 8 with 16 threads, which means that you can execute up tp 8 times faster than the single thread approach.

~

Require a software license

• In Prode Properties Tests application enter 2 to open Prode Editor

• the Editor will show the License tab, copy the ID or installation code (to copy a value select the cell and use the right button on the mouse) in this example the ID is 7J292T7H27779A3M

Rel 1.28.3 LICENSE NOT INSTALLED : CONTACT PRODE ×										
Feeds	Streams	Config	Che	mi	BIPs	Models	License			
	Installation	code			7J292	T7H27779A3M	I			
	License k	ey								
			Activate	e License	e					

• Contact Prode to receive a software license : email us the installation code and the application (commercial use or nonprofit educational institution)

Activate a software license

After the order you will receive a license key

• Open Prode Editor, select the Config tab and enter the License Key, you can also copy / paste the value : select the cell with the mouse and use the right button.

Rel 1.28.3 LICENSE NOT INSTALLED : CONTACT PRODE ×										
Feeds	Streams	Config	Cher	mi	BIPs	Models	License			
	Installation	code			7J292	T7H27779A3M				
	License k	ey		F9Y8R	7D559B3V93	8B799A4P7J2P	2R5Y5H497U			
Activate License										

• Press the button Activate License, the software will report "License Active"

P Rel 1.28.3 License 2490 support ends 20221230 user acme company										
Feeds	Streams	Config	Cher	ni	BIPs	Models	License			
	Installation	code			Lie	cense Active				
	License k	ey								
Activate License										

Prode Properties : initial setup

This section provides important information about Prode Properties initial settings.

Locating the files

The installation procedure creates different folders for program files and data files

Program files folder (Windows version) C:\Program Files\Prode\

Sample files folders (Windows version)

\Prode\C	includes support files for C / C++ applications
\Prode\Excel	includes support files for Microsoft Excel
\Prode\LIB	includes Prode Properties library files
\Prode\LibreOffice	includes support files for LibreOffice applications
\Prode\MATLAB	includes support files for MATLAB applications
\Prode\NET	includes support files for NET applications
\Prode\Python	includes support files for Python applications

Data files folder (Windows version) C:\ProgramData\prode includes these files chema.dat chemb.dat pseudo.dat bips.dat mod.dat def.ppp res.lan cfg.dat

do not remove or rename these files, when the software cannot access these files (for example because they have been disseminated in different directories) an error message "Corrupted file, error reading data file" will be generated.

Make sure all users can access data files folder

When installing Prode Properties for users without full administrative rights make sure all users have read/write rights to data files folder, if a user has no read / write rights on data files folder the program can generate errors and stop working.

Avoid errors in read / write operations

If a user doesn't receive full read / write permissions on data files folder the program can generate a error when saving def.ppp or chem.dat files,

if you see this error you can

1) login as admin, and run Prode Properties

2) immediately before to save def.ppp or chem.dat (from Prode Properties), with Windows File Manager manually delete the file which you wish to overwrite (def.ppp or chem.dat)

3) (from Prode Properties) save the file

Getting Started from Microsoft Excel (Windows version)

Prerequisites : a recent version of Excel

• different versions (32 or 64 bit) of Excel require different versions (32 or 64 bit) of Prode Properties library, by default Prode installer includes both versions.

• Take care to define commas as list separator (in Windows : open Control Panel and Region Item, then Format Tabs and Additional Settings, in List Separator select commas,)

by default Excel adopts commas to separate parameters, for example

= EStrGD(1,300,1.0E5)

Install Prode Properties add-in

• before to use Excel you must load the add-in (file properties.xla) which instructs Excel about the methods included in Prode Properties library, you need to go through this procedure when installing a new copy of Prode Properties. To install the add-in, in Excel open File menu, choose Options item which shows the Excel Options dialog



• on the bottom select Manage Excel Add-Ins and click Go, you'll see a list of add-ins, some checked, some not checked. Make sure Prode Properties isn't listed / checked, in case it is listed and checked you must remove the link to the old Add-In before to install the new.

To remove the link to the old Add-In deselect Properties, click Ok to exit the dialogs and proceed to stop Excel, then restart Excel and make sure the Add-In is not selected.

To load a new Add-In select (Browse button) the correct properties.xla file which by default is installed in C:\Program Files\Prode\Excel\properties.xla



Now Prode Properties should be listed in the list of add-ins, its box should be checked,

Close dialogs with Ok button and proceed to stop Excel

Restart Excel, a menu for Prode Properties should be visible under Add-Ins tab in Microsoft Excel

File	Home	Insert	Page Layout	Formulas	Data	Review	View	Developer	Add-ins	Help
Prop	erties 🔻									
Ed	it Properties									
Menu Coi	mmands									

• Edit Properties : to open Prode Editor

Getting Started from Microsoft Excel (predefined examples)

Prode Properties distribution includes several Excel examples to show how the software can solve a series of common problems, Excel support files are located in C:/Program Files/Prode/Excel folder.

These pages include Excel VBA code accessing Prode library, you can inspect and edit / modify the code with Excel developer tools, in the same way you can create your own custom pages.

Note : in predefined examples do not enter (in Excel cells) macros accessing Prode library to avoid conflicts and errors such as Excel not responding, to work with Excel macros see the paragraph Getting Started from Microsoft Excel (working with macros).

The first example shows how to utilize a predefined page to calculate a vapor-liquid-liquid diagram (phase envelope).

From Excel open the file phasenv.xls , define 11 as stream, click on compute phase diagram button to show the calculated results.

* : A V J# 132.721290330449	Rel 1.28.53 License 4000	support ends 2024121	12 user PRODE					×
APOR-LIQUID-SOLID PHASE DIAGRAM	Feeds	Streams	Config	Chemica	s	BIPs	Models	
roperties editor define (for the specified stream) the composition, models, BIPs (for mixtures), ct Multiphase Vapor-Liquid-Liquid or Vapor-Liquid-Solid for calculating multiphase diagrams	Operation to s	olve	T-P Flash		\sim	c	Compute	
Stream 11 compute phase diagram	Feed(s)		11		~	1 Test Case 1		
Phase Fract. 0.30	Product		11		\sim	Connect product to	feed	~
	Spec. (IN)		132.721	к	~	389978.09	Pa.a	~
	Spec. (OUT)						
000000.000000	Select Stream to	show	11		~	Flows (mole)		~
	Stream Opera	ting	132.721	к		389978.09	Pala	
0000000.000000	Phase	Feed		Vapor		Liquid	Liquid	
	Flow (kmol/s)	0.04684052	7	2.21513e-08		0.034324142	0.0125163	63
000000.000000	Energy (kW)	3175.3071	t i	0.0022436057		1382.1135	1793.191	4
	Fractions (molar)	1		4.7290883e-07		0.73278725	0.2672122	5
3000000.000000	H2O	0.73278725	5	1.0595348e-13		1	1.9840989e	-12
	N2	0.002878016	м	0.28060481		2.5443771e-14	0.0107700	28
1000000.000000	CO2	0.01106929	4	0.0014005666		1.1667297e-10	0.0414250	91
	CH4	0.14611468	3	0.71731901		6.179276e-14	0.5468099	7
000000 000000	C2H6	0.02988709	3	0.00065924087		0	0.1118477	5
	СЗНВ	0.03066194	4	1.5756691e-05		0	0.1147475	a -

for this mixture including a large fraction of Water the graph shows vapor-liquid-liquid (black) and vapor-liquid (red) lines. From Prode Editor, Stream tab, it is possible to solve multi-phase flash operations at t, p conditions predicted by procedure to generate the phase diagram (to validate results), it is also possible edit / modify the compositions, the models, the units of measurement etc.

The second example shows how to obtain a graph with hydrate formation pressures.

From Excel open the file phasenv.xls , define 6 as stream, click on compute formation curve button to show the calculated results, as for previous example you can edit / modify compositions, models, units from Prode Editor and solve multi-phase flash operations to validate results.

B C D E F G H I J K	L	Rel 1.28.53 License 4000	Streams	Config	Chemical	s BIPs	Models
CALE FORMATION CORVE rities editor define the composition including at least one hydrate former (C1, C2, C3, IC4, nC4, N2, C02, If the models support S1, SII, SH structures, you can specify different inhibitors as Methanol, Ethanol, Ethanol,	I2S,) and water a e glycol plus salts (i	Operation to Feed(s)	solve	T-P Flash 6 Test Hydrate		 ✓ 1 Test Case 1 	Compute
min 250 K Compute Hydrate Formation Curve		Product		6 Test Hydrate		✓ Connect produc	t to feed \sim
todel 1 1 = SI, SII, SH 2 = SI 3 = SII tream 6 Errors No errors		Spec. (IN Spec. (OU) ת	270	ĸ	 ✓ 18.71 ✓ 	bar.a 🗸
00 250.00 255.00 260.00 285.00 270.00 275.00 280.00 285.00 290.00 295.00	Hydrate Forma K bara	Select Stream t	o show	6 Test Hydrate		V Flows (mole)	~
	250.00 3.33 251.82 3.59 253.64 3.87 255.45 4.17	Stream Open Phase	ating Fee	270	K Vapor	18.71 Liquid	bara 🗸
	257.27 4.49 259.09 4.85 260.91 5.24	Flow (kmol/s) Energy (kW)	0.0558	52057 3826	0.055544982 5448.279	0.00030660533 20.077065	4.7003579e-0 0.026465062
	262.73 5.69 264.55 6.85 265.36 9.61 268.18 13.45	Fractions (molar) CH4	0.90	375	0.99450199	0.0054895977 0.00017955156	8.4157292e-0 0.065072242
	270 00 18 70 271.82 26.10 273 64 37 31	C2H6	0.0	15	0.050276229	2.3559958e-06	0.021038473
	275.45 55.43 277.27 99.96	coz	0.0	12	0.020110514	5.3248585#-06	0.0028797053
	279 09 187 89 280.91 263 73 282 73 363 92 284 55 485 79	H20 CH40	0.0	125	0.00026186431	0.86200631	0.8920871 7.4249985e-1
	204.00 480.72 286.36 642.05 288.18 800.23						

■ ■ = + 100% _ 10

Getting Started from Microsoft Excel (predefined examples, continuation)

A word of warning

when active, Prode Editor prevents Excel to process user inputs, close Prode Editor before to access any cell / command in Excel (to avoid possible instabilities and errors).

The third example shows how to solve mass / heat balance and calculate process properties on both sides of a heat exchanger.





on the different sides there are the buttons Edit to access Prode Editor and define / edit the compositions, the models, the units of measurement, settings etc. in addition, Prode Editor allows to solve directly many different flash operations. Click the button "Calculate Properties" to calculate temperatures and fluid properties in the different zones of heat exchanger

The fourth example shows how to solve a distillation column, from Excel open the file column.xls

🔄 🖓 - 🖓 - 🤟 columnats (Read-Only) (Compatibility Mode) - Excel 🥠 Search						×
File Home Insert Page Layout Formulas Data Review View Developer Add-ins Help					ය Shar	are
B3 → 1 × ✓ f 50	P Rel 1.28.53 Lici	ense 4000 support ends 2024	1212 user PRODE		×	*
A B C D E F G H I	J P Feeds	Streams Cor	ifig Chemicals	BIPs	Models	*
2 Number of stages 50 2. 100 stages 4 Top stage pressure 500000.00 To edit the Units access Properties Editor		Select feed	5 COLUMN FEED	1	~	
5 Boltom stage pressure 530000.00 To edit the Units access Properties Editor		Feed name	8	COLUMN FEED 1		
7 Number of feeds 1 Feed 1 Feed 2 Feed 3 Select the button to define feed comp	osition	Flow units	Molar flow		~	
Peed stage Feed stage F	edure calculates teq at	Flow (stream)	0.016526998	kmol/s	~	
10 Feeding temperature 0 To edit the Units access Properties E	ditor	Reaction set	Reaction set 1		~	
12 Number of Side Streams 0		Action	Balan	ce Chemical Equations	5	
Is a sure sure surge D	Bala	anced chemical equation				
16 17 Variable 11 0 = not present 1 = reboiler 2 = total condenser 3 = partial condenser		Chemical	ABIETIC ACID		C20F 🗸	
18 Specification type 3 1 = reflux ratio 2 = ratio top to feed 3 = ratio bottom to feed 4 = comp fm 19 Required value 0.5	act in top 5 = comp fra	Sorting criteria	Sort by first name		~	
Component Component O component or component for specifications 4,5,6,7	Add con	mponent I	Remove component	Clear list	L	
22 Variable 2 0 = not present 1 = reboiler 2 = total condenser 3 = partial condenser 23 Specification type 1 1 = reflux ratio 2 = ratio top to feed 3 = ratio bottom to feed 4 = comp fm	act in top 5 = comp fra Comp	onent	Reaction set 1	Molar fracti	ion	
25 Component 0 component position in list of components (for specifications 4 , 5 , 6 , 7	ETH	ANE No	\sim	0.0812324	493	
26 27 Solve Column Results . a numerical solution was found, please	verify the results	PANE No	\sim	0.3025210	01	
28 29 Error mass and energy balance 3 914E-16	ISOBU	JTANE No	\sim	0.0840336	513	
30 Reboiler duty 389.21265 kW	n-BU	TANE No	~	0.214752	57	
31 Condenser duty 307 7 3864 kw	n-HE	XANE No	~	0.317460	32	
33 Stage temperatures, pressures, liquid and vapor flows in K/Pa 34 T (K) P (Pa,a) LIQUID C2H6 C3H8 C4H10 C4H	110 C6H14		~	0	_	
35 Bottom product 361.70848 530000 0.008263 0 8.26E-12 8.84E-05 0.00	2928 0.005247					v b
Ready Re			III (1	I U	I + 100	00%

The page allow to define the type of column, the operating conditions, the feeds and the specifications, click on Solve Column button to solve the column and show operating conditions and compositions on the different trays.

Getting Started from Microsoft Excel (predefined examples, continuation)

The fifth example shows how to solve a polytropic stage (centrifugal compressor) with different models also including phase equilibria.

	rvcei oh							
89-		compressouds (Read-Only) (Compatibility Mode) - Excel				- 10		
File Hor	ne Insert	Page Layout Formulas Data Review View Developer Add-ins Help					ය Share	e
62		4.6.2	P Rel 1.28.53 License 4000 s	upport ends 2	0241212 user PRODE		×	
CS .		ν μ Z	Feeds Streams	Config	Chemicals	BIPs	Models	1
A	В							Ê
	pressor	/ Expander (rigorous polytropic stage) design and rating for gas and gas	Select feed		2 Test Case 2		~	-
3	Stream	Prom Properties editor define the composition for the specified stream						
4	Pin	1000000.000 Pa.a Rate in this page enter pressures, temperatures, flow (with proper units) and	Feed name		-	Test Case 2		
5	Tin	300.000 K then click on "Rate" button to estimate the polytropic efficiency, head an	Flow units		Molar flow		\sim	
7	Tout	370.000 K Result No errors	Flow (stream)		0.062170181	kmol/s	~	
8	Flow spec.	1 1 = mass flow 2 = volumetric flow (at inlet condition)			Peorting set 1			
10	Method	2 2 = Polytropic solution for gas phase (integration of 4 polos Z function) 4 = Polytropic solution with phase equilibria	Reaction set		Reaction set 1		Ť	
11			Action		Balance	Chemical Equati	ons	
12	Efficiency	0.74076 117 91968 k.lkm	Balanced chemical en	uation				
14	Power	159.18657 kW					019	
15	O days a week	INSTRUCTION	Chemical		ABIETIC ACID		~	1
17	Pin	1000000.000 Design in this page enter pressures, temperature, efficiency, flow (with proper u	Sorting criteria	6	Sort by first nam	e	~	2
18	Tin	300.000 then click on "Design" button to estimate the outlet temperature, head as						
19 20	Pout	200000.000 0.750 (0-1) Result	Add component	Remove	component	Clear li	st	18
21	Flow spec.	1 1 = mass flow 2 = volumetric flow (at inlet condition)	Component	Read	tion set 1	Molar fra	tion	đ
22	Flow	1.000				0.00		1
24	Method	1) - Polytopic solution for gas priose timegration of 4 poles 2 function) - 5 - Polytopic solution with phase equilibria	METHANE	IND	Y	0.99	1	-1
25	-		n-BUTANE	No	~	0.00	1	
20	Head				\sim	0		
28	Power				~	0		1
29					~			1
31								-1
32					~	0		4
34								
	Foglio1						•	5
Ready 💽				⊞	• •	1	- + 100	135

this page allows to design and rate polytropic stages (centrifugal compressors) also including phase equilibria (two phases operations), the procedure allows to select different methods.

The sixth example shows how to design / rate a PSV (Pressure Safety Valve) with different models as HEM, HNE, HNE-DS, NHNE.

From Excel open the file nozzle.xls Insert Page Layout Formulas Data Review View Developer Add-ins Help ය Share File Home fr B6 5 P Rel 1.28.53 License 4000 support ends 20241212 user PRODE × 1 A B C D E F G H I J K SI Co C Mod R Sizing of relief valves (critical and two-phase flow) with isentropic nozzle model 1 From Properties editor define (for the specified stream) the composition, models, BIPs (for mixtures), define discharging temperature, pressures (at inlet and outlet) and flow with proper units (you may change the select the most suitable model (1 = HEM, 2 = HNE, 3 = HNE-DS, 4 = NHNE) and the parameter (when required) 2 on set 1 Molar fraction No 0.081232493 ETHANE
 5
 the procedure estimates the (maximum, isentropic) nozzle flux and refurms the required area

 6
 Stream

 7
 Model

 1
 1 = HEM, 2 = HNE, 3 = HNE-DS, 8

 8
 Model parameter

 0
 0.7500
 No 0.30252101 1 = HEM, 2 = HNE, 3 = HNE-DS, 4 = NHNE No 0.084033613 model parameter as defined in operating m Pin Tin Pout Flow 2.000E+06 3.400E+02 1.013E+05 No 0.21475257 9 10 11 12 Pa.a UTAN K Pa.a Calculate solution No 0.31746032 n-HEXANE 1.2300 kg/s 0.3-1 0 Corrections Ka*Kb*K... 0.9000 Result : No errors 13 14 15 0 Estimated tout Calculated area 273.6703 16 17 0 4.811E-05 5.346E-05 Required Area 0 18 19 You can co of HEM n cified inlet pressure (pout set at a fraction of pin to evaluate the critical flu Stream Model to compare
 20
 Street

 21
 Mod

 22
 Mod

 23
 Pin

 24
 24

 25
 2

 26
 2

 27
 2

 30
 Mod

 31
 See

 33
 34
 1 SRK(VDW) SRK(VDW) 2 = HNE, 3 = HNE-DS, 4 = NHNE Model parameter 0.7500 Compare models 2.000E+06 Liquid Vapor GERG2008-AGA2017 V GERG2008-AGA2017 Compare HEM (Homogeneous Equilibrium) and Non Equilibrium Models GERG2008-AGA2017 GERG2008-AGA2017 20000 Quality HEM HNE 0.00025 8312.5198 18453.465 18000 GERG2008-AGA2017 GERG2008-AGA2017 16000 4000 \$2000 \$2000 0.00062 8307.81 8296.07 18374.4 18207.2 V GERG2008-AGA2017 GERG2008-AGA2017 0.00391 8266.83 17857.1 0.00977 8198.19 17146.768 Multiphase vapor-liquid-liquid 0.00977 8195.19 1740.106 0.0244141 8021.5702 15809.373 0.0610352 7627.4257 13624.304 100000 P **H** 四 8 the page allows also to compare the different models at specific condtions

Getting Started from Microsoft Excel (working with macros)

Open a new Excel page, to avoid conflicts and errors such as Excel not responding do not open / run the predefined pages when you utilize macros in Excel.

Working with Excel you can utilize Prode Editor to edit / define streams and units of measurement, this example utilizes the predefined stream 1 (Methane 0.7, Carbon Dioxide 0.15, Hydrogen Sulfide 0.15) with units Kelvin for temperature and Bar.a for pressure.

The example shows how to calculate different properties directly in Excel, we utilize the methods discussed in paragraph "Extended methods for accessing stream's properties", these methods allows to calculate properties at specified conditions, you may wish to read the paragraph for additional information.

In B1 we enter 230 as temperature (remember we have K as unit) and in B2 we enter 25 as pressure (remember we have Bar.a as unit), the units of calculated values are Kg/m3 for density, and Kj Kg / K for heat capacity

in B3 enter the macro =EStrLf(8,B1,B2) for calculating liquid fraction of stream 8 at temperature specified in B1 and pressure specified in B2

in B4 enter the macro =EStrLD(8,B1,B2) for calculating density of liquid fraction,

in B5 enter the macro =EStrLcp(8,B1,B2) for calculating heat capacity of liquid fraction,

in B6 enter the macro =EStrGD(8,B1,B2) for calculating density of vapor fraction,

in B7 enter the macro =EStrGcp(8,B1,B2) for calculating heat capacity of vapor fraction.

B3			Rel 1.28.1 License 2490 support ends 20221230 user acme company							
2	A	В	с	D	Feeds	Streams	Config	Chemicals	BIPs	Models
1	temperature	230						Sectors.		
2	pressure	25				Pressure		bar.a		
3	liquid fraction	0.076255114				Pressure (dp)		bar		
4	liquid density	919.8871524						Do		
5	liquid heat capacity	2.02329796				Temperature		ra		
6	vapor density	33.78520033			1.00	Temperature (dt)		КРа		
7	vapor heat capacity	1.999383241						MPa		
8						Calorific Value		mbar		
9					Ca	lorific Value (mola	r)	mbar		
10								bar		
11	-				E	inthalpy (Streams)		kgf/cmq		
12					-	Entropy (Streams)		psi		
13										
14					-	Heat Capacity		mmH2O		
15					He	at Capacity (mola	r)	inH2O		
17								mmHG		
18					Ma	x number of stream	ms	atm		
19					1			aun		24
20					Maxo	components per st	ream			
21		1			Max intera	ction coefficents p	er stream		100	
22					1			288 15		
23					Reference ter	nperature (normal	conditions)	200.15	ĸ	
24		-15			Reference	pressure (normal c	onditions)	1.0133e+05	Pa.a	
25										
26					Base	value for enthalpy	calc.	Specified value and t	temperature	
27					Base te	emperature for ent	halpy	1	к	
	4 1	(+)	10		Bas	e value for enthal	ру	5000	kJ/kg	1

In addition to the specific methods discussed in paragraph "Extended methods for accessing stream's properties", with Excel you can utilize all the methods exported by Prode Properties library, the list includes methods to define streams, calculate a complete set of properties and solve complex operations such as columns, reactors etc.

For exaple, you can set 150 K and 5 bar.a as operating conditions in stream 1 with the macro

=setOp(1,150,5)

in the same way you can, for example, simulate a heat exchanger (100 KW) by calculating the enthalpy of a stream to define the new operating conditions as the result of a H-P operation, where you specify 5 Bar.a as final pressure and initial enthalpy + 100 KW

= HPF(1,5,StrH(1)+100,0)

Getting started from LibreOffice

Prerequisites : a recent version of LibreOffice or OpenOffice (64 bit versions are, in general, more stable)

LibreOffice (and OpenOffice) Calc tools provide many fundamental features of Excel and they include Apache Open Office Basic, a programming language similar to Microsoft Excel VBA, Prode distribution includes several LibreOffice pages in folder /Prode/LibreOffice, the LibreOffice pages look (and work) not much differently from equivalent Excel versions, to open the LibreOffice pages. start LibreOffice Calc and Open the page phasenv.xls

) Open								×
🛧 📑 > This PC	> Local Disk	(C:) > Program Files > Prode > L	ibreoffice	~ Ö	,P Search Li	breoffice		
Organize 🔻 New folder						[iii •		0
Prode	^	Name	Date modified	Туре	Size			
C C		ase.xls	1/15/2020 1:21 AM	Microsoft Excel 97	15 KB			
Excel		compressor.xls	1/15/2020 1:25 AM	Microsoft Excel 97	26 KB			
LIB		interrops.xis	1/15/2020 8:57 AM	Microsoft Excel 97	31.KB			
Libreoffice		in hydratexis	1/15/2020 1:26 AM	Microsoft Excel 97	24 KB			
NET		🖬 multiphase.xls	1/15/2020 1:28 AM	Microsoft Excel 97	33 KB			
Buthon		nozzle.xls	1/15/2020 1:30 AM	Microsoft Excel 97	27 KB			
- vyulou		phasenv.xls	1/15/2020 1:31 AM	Microsoft Excel 97	32 KB			
system		🖬 props.xls	1/15/2020 1:34 AM	Microsoft Excel 97	39 KB			
Python		🖬 tpay.xls	1/15/2020 1:37 AM	Microsoft Excel 97	27 KB			
DLLs								
Doc								
include								
	Read-only							
File name:	phasenvixls				All files (*.*)			~
			Version	~	Onen		Cancel	ī.

you may receive a message "LibreOffice Security Warning" : The document contains document macros , click Enable Macros button and proceed

the page includes several buttons

- Properties Editor opens the editor dialog
- Open Archive opens a archive
- Save Archive saves a archive
- · Compute phase diagram calculates the phase diagram for the specified stream

To calculate the phase diagram define a stream and click the button Compute phase diagram

in the same way you can load the pages for solving different problems as discussed in Excel section

$\sim f_{\rm X} \sum \tau = 1$					Suprates - Bernin	10.000	Weaton creation	100517	
A B C D E F G H I	J K L	M	N	0	Feeds Stream	s Config	Chemic	BIPs	Mod
Properties Editor					Select fe	ed	1 Test Case 1		
(c) in Properties editor define (for the specified stream) the composition, models, BIPS (for mix c) Multiphase Vapor-Liquid-Liquid or Vapor-Liquid-Solid for calculating multiphase diagrams.	dures), Result Pc (1)	All stable poir 8306519 284	Pa a	d					_
	Tc (1)	235.390	ĸ		Feed nar	NE		Test Case 1	
Stream 1 compute phase diagram	Pc (2)				Flow un	ta	Molar flow		
Phase Fract. 0.30	Tc (2)	268.447	4		Flow (stre	am)	0.043584919	kmol/s	
	CricoP	256,447	Paa		Burden		Reading and 1		
	Surger.	Latertain	10000		Reaction	wet	Reaction set 1		_
	PHASE	DIAGRAM			Action		Balance G	Chemical Equati	ions
1000000	BP-DP		BP-DP		Balanced chemic	d equation			
	VAR:SUS K	Paa	K	Paa					_
9000000		100.0			Chemic	al Constanting	ABIETIC ACID		
8000000	112 650020	101327.400000	111.832793	101327.400	Sorting cri	teria	Sort by first nam	ne	
	121.011331	187323 454243	120.136599	188703.782					
7000000	126 011331	258833 472195	125 136595	262569.514	Add component	Remov	e component	Clear I	list
	136 011331	456931 591256	130 136599	468447 731					
6000000	141.011331	587673.852309	140.136599	604252.380	Component	Read	tion set 1	Molar fra	ction
	146.011331	742191.763396	145.136599	764053.636	METHANE	No	~	0.7	1
5000000	151.011331	922349.913962	150.136599	949100.560		110		0.1	
1000000	156.011331	1366450 668191	155.136599	1399189.27	CARBON DIOXIDE	THE	-	Q.1.	-
400000	166 011331	1633480.557451	165 136599	1666258.72	HYDROGEN SULFIDE	No	×.	0.1	5
3000000	171.011331	1932287 768857	170 136599	1962617-30			~	0	
	101.011331	2629417.744129	100 136599	2645492.46			~	0	
2000000	186 011331	3029181 266634	185.136599	3032031.63		-		U	
	193.927997	3732606.216539	190.969933	3519940.67			~	0	
1000000	201 427997 203 329936	4676626 753256	202 219933	4202756.85	tota and an even				_
	205 192540	4876626 753256			250 317827 41165	89 529439			
	207 019996	5076626 753256			251 243976 43165	89 529439			
0 100 00 120 00 140 00 160 00 180 00 200 00 220 00 240 00 26	30 00 280 00	hand a state of the state of the							

Caution : some versions of LibreOffice Calc can result unstable when loading external libraries such as Prode Editor, we observed occasional crashes when opening / closing Prode Editor and solving worksheets without pausing between operations, in such cases we recommend to pause between operations and limit access to Prode Editor. Feel free to contact Prode for specific information and support.

Getting started from Python

Prerequisites : a recent version of Python (Python 3.12 or following)

The names of methods exported by Prode Properties library via Python Add-in are listed as sname, for example xsld

Install the Python Add-in / Plug-in

• You must verify which version XXX of Python you have : the Python Add-in must match the version of Python (3.12, 3.13, ...) and type (32 / 64 bit) installed in your computer.



Python Add-ins / Plug-ins are located under C:\Program Files\Prode\Python\XXX , you can contact Prode to obtain Add-ins / Plug-ins not included with distribution.

• Copy prode.py (from C:\Program Files\Prode\Python\ XXX) to your Python install in /Lib folder

ڬ C\Program Files\Python\Lib X +						- 0 ×
$\leftrightarrow \rightarrow \uparrow \circ \Box \rightarrow ThisPo$	C > Local Disk (C:) > 1	Program Files > Python	> Lib >		Search Lib	٩
⊕ New - 🐰 🗘 🚯 8	🖞 👘 🚯 Sort -	≡ View · · · ·				D Preview
DILS	Name	Date modified	Туре	Size		
> 🖿 Doc	🗋 plistlib.py	10/2/2023 1:25 PM	Pr rile	29 KB		
> 🛅 include	poplib.py	10/2/2023 1.25 PM	Py File	15 KB		
> 10	posixpath.py	10/2/2023 1:25 PM	PY File	18 KB		
🔤 libs	C pprint.py	10/2/2023 1:25 PM	PY File	25 KB		
Scripts	prode.py	11/17/2024 9:15 AM	PY File	24 KB		
19 items						

• Copy prode.pyd (from C:\Program Files\Prode\Python\XXX) to your Python install in /DLLs folder

+ → ↑ C □ →	This PC	> Local Disk (C:) >	Program Files > Python	≻ Lib >		Search Lib	Q,
€ New - 👗 💷 🗋		🛞 🐴 Sort -	≡ View · ···				Preview
Dills		Name	Date modified	Туре	Size		
> 🛅 Doc		D plistlib.py	10/2/2023 1:25 PM	Prrue	29.68		
> 🚞 include		poplib.py	10/2/2023 1:25 PM	PY File	15 KB		
> 💴 L0	1	posixpath.py	10/2/2023 1:25 PM	PY File	18 KB		
🚞 libs		pprint.py	10/2/2023 1:25 PM	PY File	25 KB		
Scripts		prode.py	11/17/2024 9:15 AM	PY File	24 KB		
i items							

run Python shell, to import prode module in Python, type

>>> import prode

then you can access the methods exposed by Prode library, for example solve a flash at specified t, p and obtain the gas density at 230 K and 2200000 Pa (22 Bar.a) for stream 1 enter



the names to utilize with Python are those indicated in the documentation with the prefix sname

>>> prode.xsld(1) returns liquid density

>>> prode.xsgcp(1) returns gas heat capacity at constant pressure

- >>> prode.xsgcv(1) returns gas heat capacity at constant volume
- >>> prode.xslcp(1) returns liquid heat capacity at constant pressure
- >>> prode.xslcv(1) returns liquid heat capacity at constant volume

In this way you can solve unit operations and obtain fluid properties from Python Shell

Getting started from Python (continuation)

With Prode Editor you can define / edit / modify mixtures, change Units of Measurement, solve Unit Operations, access Prode Archives etc.

Pythan 3.12 (64-bit) × + ~						o x		
Python 3.12.0 (tags/v3.12.0:0fb18b0, Oct 2 2023, 13:03:3 Type "help", "copyright", "credits" or "license" for more	9) [MSC v.1935 64 information.	bit (AMD64)] on win32					
>>> import prode	P Rel 1.28.53 License 4000	support ends 202412	12 user PRODE					×
>>> prode.xedss()	Feeds	Streams	Config	Chemic	als	BIPs	Models	s
	Operation to	solve	T-P Flash		\sim		Compute	
	Feed(s)		6 Test Hydrate		\sim	1 Test Case 1		
	Product		6 Test Hydrate		~	Connect produc	t to feed	~
	Spec. (IN	0	270	к	\sim	25	bar.a	\sim
	Spec. (OU	T)						
	Select Stream t	o show	6 Test Hydrate		\sim	Flows (mole)	102	\sim
	Stream Oper	ating	270	κ		2500000	Pala	
	Phase	Feed	Va	ipor	2	iquid	Hydrate	
	Flow (kmol/s)	0.05585209	57 0.05	5532581	0.00	025651594	6.2960029e-05	
	Energy (kW)	5458.6819	9 54	37.634	17	7.504008	3.5438775	
	Fractions (molar)	1	0.99	427996	0.00	145927751	0.0011272643	
	CH4	0.90375	0.90	887139	0.00	029089596	0.067467522	
	C2H6	0.05	0.050	0263885	3.9	76313e-06	0.020943133	
	СЗНВ	0.02	0.020	0094395	3.25	08236e-07	0.018225018	
	CO2	0.02	0.020	0111575	8.30	07035e-06	0.0030389757	
	H2O	0.005	0.000	19867516	0.8	2713157	0.89032535	
	CH40	0.00125	0.0004	46007978	0.1	7256492	5.3959317e-14	

The library allows to solve unit operations as centrifugal compressors (rigorous solution of polytropic stages including phase equilibria), safety valves (with HEM, HNE, HNE-DS... models), pipes etc.



Also you can run Python scripts, note that both Python and Prode Properties support multiple threads, (concurrent execution of multiple threads within a single process), Prode distribution includes examples to compare the total time required to calculate the same series of hydrate formation pressure points in case of single and multiple threads.



In the same way you can create graphs, optimize process units, identify optimal solutions etc.

Getting started from MATLAB

Prerequisites : a recent version of MATLAB

MATLAB can access the methods in Prode Properties library in different ways :

- 1) Ioading Prode from MATLAB Command Window
- 2) via Python wrapper
- 3) linking Prode library to MATLAB code

Access Prode Properties methods from Matlab Command Window

run Matlab, then from Command Window load the list of methods in each library of Prode Properties

>> loadlibrary('ppp.dll','C:\Program Files\Prode\MATLAB\ppp.h')
>> loadlibrary('pppx.dll','C:\Program Files\Prode\MATLAB\pppx.h')

Now, to obtain the gas heat capacity of stream 1 at 270 K , 100000 Pa in Matlab Command Window enter :

>> calllib('ppp','EStrGCp',1,270,100000)

And hydrate formation pressure (in Pa) for stream 6, at 270 K

>> calllib('ppp','HPFORM',6,270.0,1)

And to show Prode Editor

>> calllib('pppx','edSS')

A MATLAB R2024b		D Rol 1 29 52	License 4000 sun	nort ands 20241			×
HOME PLOTS APPS	16	Net 1.20.35	cicense 4000 supj	port ends 202412	enz user PRODE		^
Image:	2	Feeds	Streams	Config	Chemicals	BIPs	Models
New New New Open Compare Script Live Script • • • Script Live Script • • • • • • • • • • • • • • • • • • •	3		Select feed		1 Test Case 1		~
FILE VARIABLE	-		Feed name			Test Case 1	
>> loadlibrary('ppp.dll','C:\Program Files\Prode\MATLAB\ppp.h')			Flow units		Molar flow		~
<pre>>> loadlibrary('pppx.dll','C:\Program Files\Prode\MATLAB\pppx.h') >> calllib('ppp','EstrGCp',1,270,100000)</pre>			Flow (stream)		0.043584919	kmol/s	~
so carried ppp / ascreep /r/r/ofreeded/			Reaction set		Reaction set 1		~
ans =			Action		Balance	Chemical Equation	ons
1.5243		Bala	nced chemical equ	ation			
>> calllib('ppp','HPFORM',6,270.0,1)			Chemical		ABIETIC ACID		× .
ans =			Sorting criteria		Sort by first name	e	~
1.8680e+06		Add co	mponent	Remove c	omponent	Clear li	st
<pre>>> calllib('pppx','edSS') fr</pre>	2	Comp	ponent	Reactio	on set 1	Molar fra	tion
/s		MET	HANE	No	~	0.7	é -
		CARBON		No	~	0.15	
		HYDROGI	EN SULFIDE	No	~	0.15	
					~	0	
					\sim	0	
					~	0	
					~	0	
					~	0	
					~	0	
Busy							

Getting started from MATLAB (continuation)

Access Prode Properties methods via Python wrapper

Notes :

- 1) you must have Python installed, to install Python see the paragraph "Getting started from Python"
- 2) with Python wrapper the names of Prode Properties methods are the same of Python

run Matlab, then from Command Window to obtain the gas heat capacity of stream 1 at 270 K, 100000 Pa enter : >> py.prode.xstpgcp(1,270,100000)

and hydrate formation pressure for stream 6, at 270 K >>> py.prode.xhydp(6,270.0,1)

to show Prode Editor >> py.prode.xedss()



In the same way you can solve a polytropic stage (centrifugal compressor) giving the initial conditions (t=330K, p=200000 Pa), final pressure (4000000 Pa), model (1) and efficiency (0.75), see the description of method PSPF() for additional information and details about the models available

>> py.prode.xftp(5,330,200000)

>> py.prode.xpolp(5,400000,1,0.75)

ans =

354.6350

Note : MATLAB displays all Python types as objects, which include a list of object properties. For numeric types, MATLAB displays the expected output value on the last line, here method xfcnr() returns the number of chemicals in std. version >> py.prode.xfcnr()

ans =

Python int with properties:

denominator: [1×1 py.int]

imag: [1×1 py.int]

numerator: [1×1 py.int]

real: [1×1 py.int]

Accessing Prode Properties library (with programming languages)

The technique for accessing the methods in Prode Properties library will depend on which programming language you use. Languages such as FORTRAN, C, C++ or Microsoft NET (VB,C) exhibit differences in parameter passing in and out of functions. This may require you to adapt your code from the examples shown here. The calling convention determines how a program makes a call and where the parameters are passed.

Prode Properties does use of standard calls, it pushes parameters on the stack, in reverse order. When accessing Properties consider :

- Prode Properties real (double) type is 8 bytes
- Prode Properties integer type is 4 bytes
- parameters are passed by value (with exception of strings which are arrays of characters)

C / C++

- · include the ppp.h, pppx.h headers
- · add ppp.lib, pppx.lib files to the list of the files in your project
- make sure you use the calling convention of ppp.h header file,
- from your code call the methods in Prode Properties library
- feel free to contact Prode for additional information and support

Fortran

add ppp.lib file to the list of the files in project and include ppp.f90 to instruct the compiler about the methods available in Prode Properties then access the methods as they were included in your code

C this procedure returns the critical temperature of a compound INTERFACE TO REAL*8 FUNCTION TC ([C,ALIAS:'CompTc'] comp) INTEGER*4 comp [VALUE] END REAL*8 tc INTEGER*4 id C define the id value here tc = TC(id)

Microsoft NET

we can provide samples for C# and VB#, feel free to contact us for additional information and support

Microsoft Excel (VBA)

see the examples provided, feel free to contact Prode for additional information and support

OpenOffice

see the examples provided, feel free to contact Prode for additional information and support

Python

see the examples provided, feel free to contact Prode for additional information and support

MATLAB

see the examples provided, feel free to contact Prode for additional information and support

Java, Javascript, PHP etc.

feel free to contact Prode for information and support

Some tips on creation of Prode Properties applications

• include access to Properties Editor, for example with method edSS() to simplify debug operations, when debugging always attempt to limit the complexity of problems and expand progressively to the full application, retesting at intervals as you expand the scope of your problem.

• ensure that units of measurement are correct / include methods to set the units.

• utilize isSDef() method to test a streams validity before accessing the stream, accessing undefined streams generates a large numbers of errors.

• utilize methods / procedures to test errors on each step, specifically for long calculation sequences.

Prode Properties : introduction

Streams

As other simulators Prode Properties adopts a structure based on streams (flows of material representing the piping) each stream includes

- -components
- -composition
- -reactions
- -models
- -flow

Prode Properties can store data for hundreds of streams.

How to define / edit streams

Prode Properties allows to define / edit streams from Prode Editor, the GUI distributed with Prode Properties, the paragraphs Getting Started from Microsoft Excel, Python, MATLAB etc. include examples about how to access Prode Editor from these tools.

It is also possible to define / update streams programmatically, see the paragraph "Methods to work with streams", as example, in C / C++

```
initS(stream); // init stream
```

```
for (pos=1;pos<=nrcomp;++pos)
{
    putCC(stream,pos,CC[pos]); // define component
    putZ(stream,pos,Z[pos]); // define fraction
}
```

setS(stream); // store stream

Unit Operations

With Prode Properties a process is modelled by solving a set of predefined Unit Operations as pumps / compressors, valves, distillation columns, reactors etc.

Each unit Operation includes

-one or more streams as feeds

-one or more streams as products

-a series of values (specifications) required to solve the unit

How to solve a Unit Operation

Prode Properties allows to solve unit operations programmatically or from Prode Editor, as example, to solve a polytropic stage (centrifugal compressor) with stream 1 as feed and product, Pout = 4 Bar.a, model = 1, efficiency = 0.75, there is the method PSPF()

t=PSPF(1, 4.0, 1, 0.75)

How to obtain fluid properties

The methods available in Prode Properties allow to calculate many properties and obtain the results directly in Excel, Python, MATLAB, MathCad or a custom application, note that Prode Properties exports the methods under different names depending from the application (C/C++, VB, C#, Python, Java, PHP ...)

=StrMSS(1)	; returns the speed of sound for stream 1 (gas plus liquid, HEM model)
=StrCBt(1)	; returns the cricodenBar temperature for stream 1
=StrRVP(13, 1)	; returns the Reid vapor pressure for stream 13 according D6377 procedure
=HPFORM(6, 270, 1)	; returns hydrate formation pressure for stream 6 at 270 K

Prode Editor : Introduction

Prode Properties includes a editor with several pages

- · Feeds, to edit / change compositions, flows, models, BIPs
- · Streams, to inspect streams, set operating conditions, solve unit operations
- Configuration, to define the units of measurement and settings
- Chemicals, to edit / change chemical's data, use data regression utility to calculate new values, add new chemicals
- BIPs, to edit / change bip's data, use data regression utility to calculate new BIPs
- Models, to edit / change model's data, add new models and chemicals

Rel 1.28.1 License 2490	support ends	s 202	21230 user acme	company		×
Feeds Streams	Config		Chemicals	BIPs	Mod	els
INTHANE	No				0.7	
CARBON DIOXIDE	No				0.15	
HYDROGEN SULFIDE	No				0.15	_
					0	
					0	
					0	
					0	
					0	
					0	
Predefined packages	1 SRK(VD)	W)		SR	K(VDW)	
			Solid		H	rdra
Fugacity		SPR	X-NRTL(P-HV)	H	PRX-NRTL(P	-HV
Enthalpy		REG	ULAR	н	RX-NRTL(P	HV
Entropy		REG	ULAR	H	PRX-NRTL(P	-H\
Volume		REG	ULAR	H	PRX-NRTL(P	-HV
Multiphase equi	ilibria		Multiphase vapo	r-liquid-liquid		
Multiphase initial	ization		Standard tests		_	
			E Cilitar Davi			

Prode Editor adopts a portable (Windows, Linux, Android, IOS) GUI based on a tabbed dialog with flickable grids.

The elements can be dragged and flicked causing the views to scroll, you can drag the view by pressing and holding a mouse button while moving the cursor, in addition there is a standard scroll bar for vertical scroll.

Unit conversion and data validation

Grids provide support for data validation and unit conversion, to convert to different units select a value in drop-down list

Spec. (IN)	260	к	12	bar.a
Spec. (OUT)		к		
		с		
Select Stream	1 Test Case 1	R	Flows (mole)	
Stream Operating	260	F	12	bar.a

Copy / paste operations

Copy / paste operations are allowed, use the mouse right button over a cell to activate this option (available for cells containing data inputs or results)

Change sorting criteria in combo selectors

Combo selectors (for the lists of chemicals) have two indexing options (sort by name or formula) and a quick access mechanism, typing a letter the list will scroll to the first matching value.

Reports for warnings and errors

Messages with warnings and errors are visible at the bottom of dialog, click on to delete

Prode Editor : Feeds page

From this page you can :

- select a stream (select / edit stream)
- edit / change name, stream's flow, the list of components and relative weights
- define reaction sets (for reactive flash operations), balance chemical equations for the different reaction sets
- define the models for different properties (Fg,H,S,V...), define the different settings, select BIPs dataset (VLE, LLE...)
- save / store the edited stream

P Rel 1.28.	3 License 2490 su	upport ends 202	221230 user acm	e comp	any X	
Feeds	Streams	Config	Chemicals	BI	Ps Models	
	Feed name		13 Oil-water			
	Select feed			Oil-v	water	
	Flow units		Molar flow			
	Flow (stream)		0.036218		kmol/s	
	Reaction set		Reaction set 1			
	Action			e Chem	ical Equations	
Balar	Balanced chemical equation					
	Chemical			WATER		
	Sorting criteria		Sort by first name			
Add co	omponent	Remove	component		Clear list	
Com	ponent	Reacti	on set 1		Molar fraction	
w	ATER	No			0.66373	
CARBO		No			0.010026	
MET	THANE	No			0.13234	
ETI	HANE	No			0.02707	
PRC	PANE	No			0.027772	
ISOB	UTANE	No			0.0079206	
n-Bl	JTANE	No			0.016443	
ISOPE	INTANE	No			0.0084219	

[•] use Select feed to select a stream, you can define stream's name, flow and units (molar or mass)

[•] select components from the list of chemicals and Add component / Remove component / Clear list to define composition, select Sorting criteria to obtain lists sorted by name or formula, with chemical's list open type the first character in your chemical to scroll to the first matching value

[•] For Reactive Flash operations you can define up to 5 reaction sets per stream, each components can be included as reactant, product or neutral (no), on each reaction set select Balance Chemical Equation button to obtain the chemical equation

Prode Editor : Feeds page (continuation)

P Rel 1.28.3	License 2490 su	upport ends 202	21230 user acm	e company	×
Feeds	Streams	Config	Chemicals	BIPs	Models
				0.01	
ISOBU	JTANE	No		0.00	79206
n-BU	TANE	No		0.01	6443
ISOPE	NTANE	No		0.00	84219
n-PEN	TANE	No		0.00	94245
Predefined	d packages	1 SRK(VDW)		SRK	(VDW)
		Va	por	Liq	uid
Fug	acity	PRX(VDW)		PRX(VDW)	
Enth	nalpy	PRX(VDW)		PRX(VDW)	
Enti	гору	PRX(VDW)		PRX(VDW)	
Vol	ume	PRX(VDW)		PRX(VDW)	
Mu	ultiphase equilil	bria	Multiphase vap	or-liquid-liquid	
Mul	tiphase initializa	ation	Standard tests		
D	etect Phase Sta	te	From Gibbs / Iso	othermal Compr.	and Liq.Dens.
Phase diagram	m, check stabilit	y against feed	Discard unstabl	e solutions	
Phase diag	ram, specified f	raction lines	End crossing ph	ase boundary	
Hydra	te structures inc	clusion	Include normal	structures gener	ated by formers
	Source for BIPs	i	Prode VLE data	set	
	Save / store feed	ł		Store in File	

• Define the models for the different properties (fugacity, enthalpy, entropy, volume) and state (vapor, liquid, solid, hydrate) you can select from the lists or use one of predefined packages

• the editor allows to edit existing packages and define new packages, select a package in the list, the models you wish to define, enter a name for the package and use button Save to store the new package

- Define settings
- Multiphase equilibria, allows to define different solutions as vapor-liquid, vapor-liquid-liquid and vapor-liquid-solid
- Multiphase initialization, allows to reduce the number of trial phases thus reducing time required
- Detect Phase State, allows to use different methods to detect the state of each phase
- Phase diagram, check stability against feed, allows to include stability analysis on each calculated point
- Phase diagram, specified phase fraction lines, allows to terminate lines when crossing a phase boundary
- Hydrates structures inclusion, allows to test all possible hydrate structures which may be generated by former(s)
 Select the source for BIPs
- Prode VLE dataset, includes a large collection of BIPS calculated from VLE data points
- Prode LLE dataset, includes a limited number of BIPS calculated from LLE data points
- Prode SLE dataset, includes a limited number of BIPS calculated from SLE data points

Prode Hydrate dataset, includes a limited number of BIPS specific for hydrate phase equilibria

- · Select Save / store feed button, the program will store your data
- Select Store in file button, the program will save your data in file

Prode Editor : Streams page

From this page you can :

• Inspect streams, solve flash operations, mix streams, solve vapor-liquid, liquid-liquid separations

P Rel 1.28.51	1License 4000 s	upport ends 202	41212 user PRC	DDE	×	
Feeds	Streams	Config	Chemicals	BIPs	Models	
Operatio	Rel 1.28.51License 4000 : Feeds Operation to solve Feed(s) Product Spec. (IN) Spec. (OUT) Select Stream to show Stream Operating Phase Flow (kmol/s) Energy (kW) Fractions (molar) CH4 CO2 H2S	T-P Flash		Con	npute	
Fee	ed(s)	T-P Flash		1 Test Case 1		
Pro	duct	PF-P Flash		Connect produc	ct to feed	
Spec	Rei 1.28.5 Elicense 4000 Feeds Streams Operation to solve Feed(s) Product Spec. (IN) Spec. (OUT) Select Stream to show Stream Operating Phase Flow (kmol/s) Energy (kW) Fractions (molar) CH4 CO2 H25	PF-T Flash		10	bar.a	
Spec.	(OUT)	H-P Flash				
-		H-T Flash				
Select Stre	Spec. (OUT) H-P Flash Select Stream to show S-P Flash Stream Operating S-T Flash Phase V-P Flash Flow (kmol/s) Copy Stream Energy (kW) Gas Separator			Flows (mole)		
Stream (1000000	Pa.a	
		V-P Flash				
Pha	ase	V-T Flash		Vapor 0.032461734 3090.6064 0.74479281		
Flow (k	cmol/s)	Copy Stream				
Energy	(kW)	Gas Separator				
Fractions	s (molar)	Liquid Separato	r	Flows (mole) 1000000 Pa.a Vapor Colspan="2">Colspan="2"Colspan=		
Cł	44	Mixer		0.92760)713	
cc	D2			0.06041	0652	
н	25	0.1	5	0.01198	2222	
		0		0		
		0		0		
		0		0		

Inspect a stream

• use Select Stream to select a stream, note that depending from selected option (connect product to feed or do not connect) the selection may change when the feed or product change

Compute flash operations

• make sure all feeding streams have been defined

• select feeding streams, product stream and the operation to solve, there is an option to connect the selected stream (and product) to feed or product (to view results)

enter the required specifications and select "Compute"

List of operations which you can solve from Prode Editor

Flash at specified Temperature and Pressure

Flash at specified Liquid Fraction and Pressure or Temperature

Flash at specified Enthalpy and Pressure or Temperature

Flash at specified Entropy and Pressure or Temperature

Flash at specified Volume and Pressure or Temperature

Copy Streams

Vapor-Liquid and Liquid-Liquid separators Mixers

**Customized versions can include additional operations

Prode Editor : Config page

From this page you can define

- units of measurement
- parameters, options and preferences (settings) utilized by Prode Properties



Setting the units of measurement

With Prode Properties you have complete control over the engineering units

• select your preferred units from the list available for each property

• select Set new configuration values button to update configuration, the program will convert automatically the input values and the results accordingly

Prode Editor : Config page (continuation)

P Rel 1.28.	3 License 2490 sup	oport ends 202	21230 user acme comp	oany X	
Feeds	Streams	Config	Chemicals B	IPs Models	
	Calorific Value		kJ/kg		
Ca	alorific Value (mola	ar)	kJ/kmol		
	Enthalpy (Streams))	kW		
	Entropy (Streams))	kJ/(K*s)		
	Heat Capacity Heat Capacity (molar)		kJ/(kg*K)		
н	eat Capacity (mola	ar)	kJ/(kmol*K)		
Base	value for enthalpy	/ calc.	Specified value and ten	nperature	
Pass		4h a laur	1	v	
Dase t	emperature for en	unaipy	5000	ĸ	
Ba	se value for entha	Іру	5000	kJ/kg	
Base	value for entropy	calc.	Specified value and ten	nperature	
Base t	temperature for er	tropy	1	κ	
Ba	se value for entro	ру	50	kJ/(kg*K)	
Convergen	ce tolerance on sp	ecifications	16	-09	
Max allowe	ed time for solving	operations	60	5	
	Flow units		Molar flows		
Minimu	ım Density for liqu	id state	200	kg/m3	
Set r	new configuration v	alues	Store	in File	

configurable parameters :

- max number of streams
- max number of components per stream
- max number of interaction coefficients pairs per stream
- reference temperature and pressure
- base values for enthalpy and entropy calc's
- convergence tolerance
- max allowed time for solving a operation
- Flow units
- minimum liquid density to validate liquid phase

• select Set new configuration values button to update configuration, the program will adopt the new configuration parameters

• select Store in file button to store actual configuration in file, the program will adopt the new configuration parameters as default values

Prode Editor : Chemicals page

From this page you can :

• Inspect / edit physical properties data stored in the software, regress raw data, add / remove components

P Rel 1.28.	3 License 2490 sup	port ends 202	221230 user acme com	pany X		
Feeds	Streams	Config	Chemicals B	IPs Models		
	Chemical		WATER			
	Sorting criteria		Sort by first name			
	Code			21		
	Formula		ŀ	120		
	Name (1)		W	ATER		
	Name (2)		w	ATER		
	Name (3)					
CAS	CAS / Identification number		7732185			
	Molecular weight		18.015			
	Critical temperature	,	647.1	к		
	Critical pressure		2.2064e+07	Pa.a		
	Critical volume		0.055948	m3/kmol		
	New Component		Remove (Component		
	Store Component		Store in File			
	Property		Vapor Heat capacity or	nuation		
	Correlation		v=a+b*t+c*t^2+d*t^	2		
	conclation		y-arb (+c (2+a (*)			

Inspect / edit data :

- select the component from the component's list
- · edit / modify the related fields
- select "Store Component" button to save the modified data
- Adding a new component :
- select "New Component" button
- edit the related fields
- select "Store Component" button to save data
- Remove a component :
- select a component from the component's list
- select "Remove Component" button
- Update the files which stores physical properties data :
- select the "Save File" button, this command overwrites the file chem.dat, if required you can create a backup

Prode Editor : Chemicals page (continuation)

P Rel 1.28.	3 License 2490 su	ipport ends 202	21230 user acm	e comp	any	×
Feeds	Streams	Config	Chemicals	BI	Ps	Models
	Critical volume		0.055948		m3/kmol	
	New Component	:	Re	emove C	omponent	
	Store Component	t		Store	in File	
	Property		Liquid density e	quation		
Correlation			y=a+b*t+c*t^2	+d*t^3	1	
-		v			C.II.	
lem	perature	Va	lue		Calcula	ted
283.1	к	55.492	kmol/m3		55.49	4
293.1	к	55.409	kmol/m3		55.40	6
303.1	к	55.267	kmol/m3		55.26	6
313.1	к	55.077	kmol/m3		55.07	'8
323.1	к	54.845	kmol/m3		54.84	7
333.1	к	54.576	kmol/m3		54.57	7
343.1	к	54.274	kmol/m3		54.27	/3
353.1	к	53.943	kmol/m3		53.94	11
363.1	к	53.583	kmol/m3		53.58	34
	Clear list			Calc	ulate	

Note : Prode Properties supports more than 15 different correlations per each property, you can select the correlation which best fits experimental data

Regress raw data

- select a chemical
- select a property and the correlation for fitting raw data
- enter the available data (all temperature and value pairs) with the proper units of measurement
- select Calculate button , the procedure adds the calculated parameters to the database
- evaluate calculated values and errors, you may try different correlations for best data fitting
- select "Store Component" button to save the new data
- Update the file which stores physical properties data :

• select "Save File" button, this command overwrites the file chem.dat, if required you can create a backup

Prode Editor : BIPs page

From this page you can :

- edit Binary Interaction Parameters
- add / remove Binary Interaction Parameters
- regress VLE (vapor-liquid) , LLE (liquid-liquid) , SLE (solid-liquid) data points
- save all data in a file

P Rel 1.28.3	License 2490 supp	oort ends 202	21230 user acme comp	oany X		
Feeds	Streams	Config	Chemicals B	Ps Models		
	Chemical		WATER			
	Chemical		ACETONE			
	Sorting criteria		Sort by first name			
	Models		NRTL			
s	elect BIPs Data Set		VLE BIPs			
м	in temp.in data set	t	329.21	к		
м	ax temp.in data set	t	370.02	к		
м	in press.in data set	t	1.0133e+05	Pa.a		
м	ax press.in data set	t	1.0133e+05	Pa.a		
Х-Ү	data fitting error 9	%	0.5	8271		
	U12		12	94.1		
	U21		46	1.79		
	A12		0.4	268		
	Store value		Store	in File		
Mo	del for vapor phas	e	REGULAR			

Edit / modify data :

- · select two components from the component's lists
- select the database (VLE/LLE/SLE/Hydrate)
- select the model
- edit / modify BIPs
- select "Store value" button to save the modified data

Update the file which stores physical properties data :

• select "Store in File" button, this command overwrites the file bips.dat

Prode Editor : BIPs page (continuation)

P Rel 1.28.3 License 2490 st	upport ends 202	221230 user acm	e company	×		
Feeds Streams	Config	Chemicals	BIPs	Models		
Store value			Store in File			
Model for vapor p	hase	PRX-NRTL(P-HV)			
Model for liquid p	hase	PRX-NRTL(P-HV)			
Model for solid pl	nase	SPRX-NRTL(P-H	v)			
Regress		measured VLE-I	LLE-SLE data poi	nts		
Bips data set to so	olve	Standard set, more accurate but slow				
Minimization mode		F = xerr * yerr				
Calculate			Clear table			
Туре	1	X1	١	(1		
VLE	0	.999	0.9	6127		
VLE	0.9	94653	0.3	4394		
VLE	0.8	89401	0.2	5409		
VLE	0.8	34148	0.2	2316		
VLE	0.7	8895	0.2	0762		
VLE	0.7	/3642	0.1	9728		
VLE	0.6	68389	0.1	8877		
VLE	0.6	53136	0.1	8076		
VLE	0.5	7883	0.1	7273		

Calculate BIPs with Data Regression Utility

Enter experimental VLE-LLE-SLE data points or generate VLE points with a predictive model

- select the Chemicals
- select BIPs Data Set
- select the Models for vapor, liquid, solid phases
- select the type of data points
- select the BIPs data set to solve
- select Minimization mode
- If you have selected Regress measured VLE-LLE-SLE data points enter one point per row,

in Type select VLE / LLE / SLE

- in X1, Y1 enter the molar fractions of first component C1 in the different phases,
 - for VLE : liquid in X1, vapor in Y1,
 - for LLE : liquid (phase 1) in X1, liquid (phase 2) in Y1,
 - for SLE : liquid in X1, solid in Y1,

enter the temperature and the pressure for that point.

• If you have selected Regress VLE points calculated with UNIFAC the procedure will calculate the required VLE points

- select Calculate button , the procedure adds the calculated BIP values to the database
- select "Store Value" button to save the new data
- select "Save File" button to save these values in bips.dat.file

Prode Editor : Models page

From this page you can :

• edit the parameters required by the different models available in library

P Rel 1.28.	3 License 2490 sup	port ends 202	221230 user acme	company	×			
Feeds	Streams	Config	Chemicals	Chemicals BIPs Mod				
	Chemical		WATER					
	Sorting criteria		Sort by first name	e				
	Models			-HV)				
	R			1.4				
	Q			0.92				
	A			5.9981				
	В			0.019106				
	c			0.14829				
	D			0.65433				
	E			2.9876				
	F			1.1415				
	G			-0.00017736				
	н			3.9695e-05				
	Save			Store in File				

Edit / modify data :

- select the component from the component's lists
- select the model
- edit / modify the parameters
- select the "Save" button to save the modified data
- select "Save File" button to save these values in mod.dat.file

Prode Editor examples : define / edit streams

This example shows how to define / edit a mixture in stream 45 with Prode Editor

Methane	0.8
Ethane	0.1
Propane	0.07
N-Butane	0.03
Model	Peng-Robinson std. (for all properties, fugacity, enthalpy, entropy, volume)

open Prode Editor, Feed tab, select stream 45, define / edit the name (test case) for this stream, define flows etc.
 from the list of components (Chemical) select the first component, Methane and click on button Add component
 from the list of components select the second component, Ethane and click on button Add component

Rel 1.28.54	License 4000 supp	oort ends 20241	212 user PRODE		×	P Rel 1.28.5	4 License 4000 sup	port ends 2024	1212 user PRODE		
Feeds	Streams	Config	Chemicals	BIPs	Models	Feeds	Streams	Config	Chemicals	BIPs	Models
	Select feed		MAGNESIUM OXIDE	MgO	-		Select feed		45		
	Feed name		MAGNESIUM SULFATE	MgO4S			Feed name			Test case	
	Flow units		MALATHION	C10H190	D6PS2		Flow units		Molar flow		
	Flow (stream)		MALEIC ACID	C4H4O4			Flow (stream)		1	kmol/s	
Reaction set		MALEIC ANHYDRIDE	C4H2O3	C4H2O3		Reaction set		Reaction set 1		×	
	Action		MALEONITRILE	C4H2N2			Action		Balance	e Chemical Equa	tions
Bala	nced chemical equa	ation	MALIC ACID	C4H6O5		Bal	anced chemical equ	ation			
			MALONIC ACID	C3H4O4			25				
Chemical		MALONONITRILE	C3H2N2		Chemical			n-BUTANE			
	Sorting criteria		MELAMINE	C3H6N6			Sorting criteria		Sort by first nam	ne	
Add co	omponent	Remove c	MERCURY	Hg		Add co	omponent	Remove	component	Clear	r list
Com	nonent	Ponctie	MESITYLENE	C9H12		Com	nonent	Ponct	ion cot 1	Malarfi	action
	ponent	neuch	MESITYL OXIDE	C6H10O			ponent	Neuco		inicial in	action .
			METHACROLEIN	C4H6O		ME	THANE	NO	~	U	.8
			METHACRYLIC ACID	C4H6O2		ET	HANE	No	~	0	.1
			METHACRYLONITRILE	C4H5N		PR	OPANE	No	~	0.	07
			METHANESULFONIC AC	CID CH4O3S		n-B	UTANE	No	\sim	0.	03
			METHANE	CH4					~		0
			METHANOL	CH4O					\sim		0
			METHOXYACETIC ACID	C3H6O3							

4) define Peng Robinson as model for all gas and liquid properties (you can also select from predefined packages list)

Rel 1.28.54 License 40	uu support ends 2024121.	2 USER PRODE				×
Feeds	Streams	Config	Chemicals	BIPs	Models	
		UNIFAC (REV.5)	~		0	
		PR(VDW)	~		0	
Predefined packag	es 2 PR(VDW)	PRX(VDW)	PR(VDW)		Save	
	Vapor	PRX-NRTL(P-HV)		Solid	Hydrate	
Fugacity PR(VDW) V	PRX-WILSON(P-H	V) SPRX-NRT	L(P-HV) \checkmark	HPRX-NRTL(P-HV)		
Enthalpy	PR(VDW) V PRX-UNIQUAC(P-HV)	- REGULAR	~	HPRX-NRTL(P-HV)		
Entropy	PR(VDW)	PRX-NRTL(WS)	REGULAR	~	HPRX-NRTL(P-HV)	12
Volume	PR(VDW)	PRX-UNIQUAC(W	5) REGULAR	×	HPRX-NRTL(P-HV)	
	Multiphase equilibria	PRX-WILSON(MH	v2) a vapor-liquid-li	quid		~
•	Aultiphase initialization	PRX-NRTL(MHV2)	PRX-NRTL(MHV2) ests			
e 1.000	Detect Phase State	PRX-UNIQUAC(M	PRX-UNIQUAC(MHV2) is / Isothermal Compr. and Liq.Dens. PRX-WILSON(P-LCVM) thresold (standard tests) PRX-NRTI (P-LCVM) po phase boundary			1
Stabulty an	arysis, check stability again	ines PRX-WILSON(P-LX				~
Ну	drate structures inclusion	PRX-UNIQUAC(P-	LCVM) irmal structures	M) rmal structures generated by formers dataset		
	Source for BIPs	BWRS	dataset			
	Save / store feed	LKP GERG2008-AGA2(17	Store in File		
		PRXCPA(VDW)				

5a) <u>click on Save button to save this stream in memory, note that this data will be lost when closing the app</u> 5b) <u>click on Save / Store in File button to save this stream in file to have it available for future work.</u>

Prode Editor examples : define / edit streams (continuation)

Note : when required Prode Properties can include in each stream up to 5 Reaction sets, in this example Reaction set 3 (combustion modeling) includes C2 and O2 as reactants and CO2 and H2O as products, click on Balance Chemical equations to solve the set

P Rel 1.28.54 License 40	00 support ends 20241212 user i	RODE				>	
Feeds	Streams	Config	Chemicals	BIPs	Models		
	Flow units		Molsr flow				
	Flow (stream)		1 kmoj/s			14	
	Reaction set		Reaction set 3				
	Action		Balance Chemical Equations				
	Balanced chemical equation		2 (C2H	6) + 7 (02)> 4 (002) + 6 (H20	2)		
	Chemical Sorting criteria		WATER Sort by first name				
Add	Add component Ren		ave component				
0	emponent	R	Reaction set 3 Molar fraction				
	WETHANE	No	~	0.5	0.5		
	ETHANE	Reactant		0.1	0.1		
10	PROPANE No 		No 0.07 No 0.03 Restant 0.3				
CAR			~	0			
	WATER	Product		0			
			~	0			

Note : Prode Properties allows to define for each stream a series of options

-for phase equilibria solve vapor+liquid / vapor+liquid+liquid / vapor+liquid+solid / vapor+liquid+solid+hydrates -define different procedures to initialize multiphase calc's

-detect phase state (vapor / liquid / solid...) with different methods

- -check stability of each phase at equilibrium with different methods
- -choose the methods to calculate phase diagrams
- -include standard or all hydrate structures when solving phase equilibria including hydrates

-include BIPs from different archives (optimized for different types of phase equilibria calculations)

Predefined packages	2 PR(VDW)		~	PR(VDW)		Save	
	Vapor		Liquid		Solid		
Fugacity	PR(VDW)	\sim	PR(VDW)	\sim	SPRX-NRTL(P-HV)	\sim	HPRX-NF
Enthalpy	PR(VDW)	\sim	PR(VDW)	\sim	REGULAR	\sim	HPRX-N
Entropy	PR(VDW)	~	PR(VDW)	~	REGULAR	\sim	HPRX-N
Volume	PR(VDW)	~	PR(VDW)	~	REGULAR	\sim	HPRX-NF
Dete Stability analysis, Phase diagram	Detect Phase State Stability analysis, check stability against feed Phase diagram, specified fraction lines			Multiphase vapor-liquid-solid Multiphase vapor-liquid-solid-hydrate			
Hydrate s	tructures inclusion		Include no	ormal structo	ures generated by for	mers	\sim
Sou	urce for BIPs		Prode VLE	dataset			\sim
Saure			Save / store in File				

Note : click on Save button to save this stream in memory or click Save / Store in File button to save this stream in file

Prode Editor examples : set units of measurement and configuration settings

This example shows how to select different Units of Measurement and settings from Prode Editor. 1) open Prode Editor, Config tab and select the preferred Units of Measurement

Feeds	Streams	Config	Chemicals	BIPs	Models
	Pressure		Pa.a		
	Pressure (dp)		Pa.a		
	Temperature		Pa.g		
	Tomporphyse (dt)		mbar.a		
	Temperature (ut)		mbar.g		
	Calorific Value		kPa.a		
	Calorific Value (molar)		kPa.g		
E	nthalpy streams (H * flor	w)	1992		
E	ntropy streams (S * flow	v)			
	Heat Capacity		barg		
	Heat Capacity (molar)		MPa.a		
			MPa.g		
	Max number of streams	E.	kgf/cmq.a		
м	ax components per stre	am	kgf/cmq.g		
Max int	teraction coefficents per	stream	psi.a		
Reference	temperature (normal c	onditions)	psi.g		
Referen	ce pressure (normal con	ditions)	atm.a		
Ba	se value for enthalpy ca	ılc.	atm.g		
Bas	e temperature for enth	lpy	mmH2O.a		
	Base value for enthalpy		mmH2O.g		
в	ase value for entropy ca	k.	inH2O.a		
P	se temperature for onte	2004	inH2O.g		

In addition to the Units of Measurement Prode Properties allows to define

- -Max number of streams
- -Max number of components per stream
- -Reference temperature and pressure at normal conditions
- -Base values when calculating enthalpy and entropy
- -Convergence tolerance
- -Flow units

-Minimum density allowed for liquid phase

Base value for enthalpy calc.	Specified value and	temperature	\sim
Base temperature for enthalpy	1	к	\sim
Base value for enthalpy	5000	kJ/kg	~
Base value for entropy calc.	Specified value and	temperature	~
Base temperature for entropy	1	к	\sim
Base value for entropy	50	kJ/(kg*K)	~
Convergence tolerance on specifications		1e-09	
Max allowed time for solving operations	60	5	\sim
Flow units	Molar flows		\sim
Minimum Density for liquid state	200	kg/m3	~
Set new configuration values		Save / store in File	

click on Set new configuration values button to set these values
 click Save / Store in File button to set these values and store the new configuration in file

Prode Editor examples : inspect streams, solve flash operations

This example shows how to inspect streams and solve flash operations from Prode Editor.

Open Prode Editor, Streams tab, to inspect a stream select the name in Feeds or Stream to Show, Prode Editor will show -Operating conditions (or default values when there are not values stored for the selected stream)

-Flow (in Mass, Moles or Volumes) of the feed and each active phase

-Energy of the feed and each active phase

-Composition of the feed and each active phase



Prode Editor allows to solve flash operations with different specifications, as example we model a control valve wih Pin 30 Bar.a, Tin 280 K and Pout (10 Bar.a), the valve is modeled as adiabatic stage (dH = 0.0) and the solver will calculate Tout, n Prode Editor, Streams tab:

- -as Operation to Solve select H-P Flash
- -as Feed select stream 4
- -as Product select stream 4 (the results will be stored in stream 4)
- -In Spec (IN) enter Tin 280 K and Pin 30 Bar.a

-In Spec (OUT) enter Pout 10 Bar.a K and dH 0.0 (no heat added to / removed from the valve)

Then click on button Solve to see the results

Rel 1.28.54 Lice	ense 4000 support e	ends 20241212 user l	PRODE				
Feeds	Streams	Config	Chemicals		BIPs	Mode	ls
Operatio	in to solve	H-P Flash	8	~	G	ompute	
Feed(s) Product Spec. (IN)		4 Test Case 4 $$ $$ $$		1 Test Case 1			
		4 Test Case 4 V Connect product to feed			ct to feed		
		290	к ~		30	bar.a	
Spec.	(OUT)	10	bar.a	~	0	kW	
Select Stre	am to show	4 Test Case 4		~	Flows (mole)		3
Stream (Operating	276.87656	К	4	10	bar.a	
Phase		Feed	Vapo	r		Liquid	
Flow (kmol/	/s)	0.048287314	0.04730	8772		0.00097854206	
Energy (kW	0	5416.3463	5350.9299 0.97973501		65.41632 1 0.020264993		1
Fractions (mo	olar)	1					
CH4		0.78	0.79613363		8.1581331e-07		
С2Н6		0.1	0.10205757		7 0.000524253		
СЗН8	C3H8 0.05		0.051034202			2.7875325e-07	
CO2		0.02	0.020413625		e l	2.8167046e-06	
C4H10		0.03	0.02952	5675	6	0.05293178	
H2O		0.02	0.000835	3000	19	0.94654006	

Prode Editor examples : edit chemicals, add new components

This example shows how to edit a chemical and add a new component.

To edit a chemical

1) open Prode Editor, Chemicals tab, from the first list select the chemical, the Editor allows to inspect / edit the values 2) click on button Save Component to save the values in memory or Save / store in File to save in File.

Data regression procedure

Prode Editor includes a data-regression procedure to calculate the parameters of selected temperature dependent correlation in order to fit a series of experimental data points, see the example

1) as chemical select Ammonia

2) as Property select Liquid density equation

3) as Correlation select $y=a+b^{(1-tr)^{0.35+c^{(1-tr)^{(2/3)+d^{(1-tr)+e^{(1-tr)^{(4/3)}}}}$

4) enter the following points (ammonia liquid densities) and click on Calculate button

 $y=a+b^{*}(1-tr)^{0.35+c^{*}(1-tr)^{(2/3)}+d^{*}(1-tr)+e^{(1-tr)}}$

Temperature (C) Saturated liquid density (Kmol / m3)

-70	42.55
-50	41.22
-30	39.799
-20	39.05
0	37.48
20	35.828
40	33.47
60	32.01

Rel 1.28.54 License 4000 support ends 20241212 user PRODE X

Feeds	Streams	Config	Chemicals	BIPS	Models	
C/	CAS / Identification number		7664417			
	Molecular weight			17.0306		
	Critical temperature		405.52499	к	~	
	Critical pressure		11290000	Pa.a	~	
	Critical volume		0.072400004	m3/kmol	~	
	New Component		Ren	nove Component		
	Save Component		Sa	ve / store in File		
	Property		Liquid density equati	ion	~	

Ter	mperature		Value	Calculated	Error %
-70	c ~	42.55	kmol/r 🗸	42.592194	0.099162311
-50	c ~	41.22	kmol/r 🗸	41.10601	-0.27654123
-30	c ~	39.799	kmol/r 🗸	39.817486	0.046449325
-20	c ~	39.05	kmol/r 🗸	39.135455	0.21883389
0	с ~	37.49	kmol/r 🗸	37.552806	0.16752827
20	c ~	35.828	kmol/r 🗸	35.637324	-0.53219714
40	c ~	33.47	kmol/r 🗸	33.586343	0.34760478
60	c ~	32.01	kmol/r 🗸	31.985485	-0.07658594
0	к ~	0	kmol/r 🗸	0	0
	Clea	r list		Calcula	te

the results show the calculated values and relative errors, the calculated parameters are visible in second grid (Liquid Density Equation).

5) <u>click on button Save Component to save the calculated values in memory or Save / store in File to save in File (for future work).</u>

To add a new chemical

1) click on button New Component

Correlation

2) enter the values required (these could include critical parameters plus a few temperature dependent correlations (as vapor heat capacity required to solve enthalpy / entropy) or more data.

3) <u>click on button Save Component to save the calculated values in memory or Save / store in File to save in File (for future work).</u>
Prode Editor examples : edit BIPs, calculate new BIPs

This example shows how to edit Binary Interaction Parameters and calculate new BIPs.

To edit Binary Interaction Parameters

- 1) open Prode Editor, BIPs tab, from the first two lists select the chemicals, for example methanol and water
- 2) select the model (for example Wilson) and BIPs data set (for example VLE BIPs)
- 3) the Editor allows to inspect / edit the values
- 4) click on button Save to save the edited values in memory or Save / store in File to save in File.

P Rel 1.28.54 License 4000 support ends 20241212 user PRODE X						
Feeds	Streams	Config	Chemic BI	Ps	Models	
8	Chemical		METHANOL		\sim	
	Chemical		WATER		\sim	
	Sorting criteria		Sort by first name		\sim	
	Models		WILSON		\sim	
S	elect BIPs Data Se	t	VLE BIPs		\sim	
N	1in temp.in data se	et	337.82999	К	\sim	
M	lax temp.in data se	et	372.56	к	\sim	
N	1in press.in data se	et	101327	Pa.a	\sim	
M	lax press.in data se	et	101327 Pa.a		\sim	
X-1	Y data fitting error	%	0.64459711			
	U12		-51.030968			
	U21		625.24664			
	U21					
	U12-T					
	U21-T					
	Save		Save / ste	ore in File		
M	odel for vapor pha	se	REGULAR		\sim	

Data regression procedure

Prode Editor includes a data-regression procedure to calculate BIPs of selected EOSs,, the procedure permits to enter experimental (measured) VLE-LLE-SLE data points or fit automatically the model to VLE points calculated with UNIFAC. The first example calculates BIPS for Methanol-Water from measured VLE points

Measured VLE data points for Methanol-Water at 735 mmHg

Point	X(1)	Y(1)	Temperature (C)	Pressure (mmHg.a)
VLE	0.008400	0.103000	96.5	735
VLE	0.025800	0.227000	92.3	735
VLE	0.068000	0.391000	87.5	735
VLE	0.137000	0.568000	80.1	735
VLE	0.240000	0.680000	75.9	735
VLE	0.480000	0.790000	70.6	735
VLE	0.572000	0.820000	68.7	735
VLE	0.741000	0.906000	66.4	735

In Prode Editor, Feed tab :

- 1) as first chemical select Methanol
- 2) as second chemical select Water
- 3) select the models for the vapor and liquid phases (in this case PRX-NRTL-HV)
- 4) select measured VLE-LLE-SLE data points as data to regress

5) select Standard set as solution mode, this option allows to calculate Gij, Gij and Aij for the specified model, you do not select the Standard set the procedure calculates only the Kij in base Peng-Robinson EOS

6) select F = xerr*yerr as minimization mode

9) enter the measured data (see previous page) selecting the proper units

10) click on calculate button

P Rel 1.28.54 License	4000 suppo	ort ends 20241212	user PRODE					×
Feeds	Str	eams	Config	Chemical	S	BIPs	Mode	els
	S	ave			Save	/ store in File		
	Model for	vapor phase		PRX-NRTL(P-H	V)			\sim
	Model for	liquid phase		PRX-NRTL(P-H	V)			\sim
	Model fo	r solid phase		SPRX-NRTL(P-I	HV)			\sim
	Re	gress		measured VLE-	LLE-SLE data po	oints		\sim
	Bips data	set to solve		Standard set, more accurate but slow				
	Minimiz	ation mode		F = xerr * yerr				\sim
	Cal	culate			C	lear table		
Туре		X1		Y1	Tempe	rature	Pres	sure
VLE	\sim	0.0084		0.103	96.5	c ~	735	mmHG
VLE	\sim	0.0258		0.227	92.3	c ~	735	mmHG
VLE	~	0.068		0.391	87.5	c ~	735	mmHC
VLE	\sim	0.137		0.568	80.1	c ~	735	mmHG
VLE	\sim	0.24		0.68	75.9	c ~	735	mmHG
VLE	\sim	0.48		0.79	70.6	c ~	735	mmHG
VLE	\sim	0.572		0.82	68.7	c ~	735	mmHG
VLE	\sim	0.741		0.906	66.4	c ~	735	mmHC
VLE	\sim	0		0	0	К ∨	0	bar.a

11) the procedure shows (on second table) the calculated parameters (U12,U21,A12) and (last table, on right of measured values) the calculated values and the relative errors

12) click on button Save to save the values calculated in memory or Save / store in File to save in File.

Feeds Streams Config Chemicals Models PRX-NRTL(P- Select BIPs Data Set VLE BIPs	s <mark>BIPs</mark> Mo	dels			
Models PRX-NRTL(P- Select BIPs Data Set VLE BIPs	P-HV)				
Select BIPs Data Set VLE BIPs		\sim			
		\sim			
Min press.in data set 735.00002	mmHG.a	\sim			
Max press.in data set 735.00002	mmHG.a	\sim			
X-Y data fitting error %	2.970376				
К12	0				
U12	9994.0156				
U21	1308.0734				
U12-T	0				
U21-T	0				
A12	0.33671153				
Save	Save / store in File				

The second example shows how to estimate BIPs from available Solid-Liquid equilibrium points

Measured SLE data points for Water-Methanol at atmospheric pressure Note : X(1) is the water fraction in liquid phase while Y(1) is solid fraction (we can set 1 for a solid pure model)

Point	X(1)	Y(1)	Temperature (K)	Pressure (Bar.g)
SLE	0.9432	1	266.85	0
SLE	0.9	1	259.65	0
SLE	0.87676	1	257.65	0
SLE	0.80583	1	246.85	0
SLE	0.8	1	242.95	0
SLE	0.728	1	233.45	0
SLE	0.641	1	217.95	0
SLE	0.636	1	214.95	0
SLE	0.6	1	208.15	0

In Prode Editor, Feed tab :

1) as first chemical select Water

2) as second chemical select Methanol

3) select the models for the vapor , liquid and solid phases (in this case PRX-NRTL-HV)

4) select measured VLE-LLE-SLE data points as data to regress

5) select Standard set as solution mode, this option allows to calculate Gij, Gij and Aij for the specified model

6) select F = xerr*yerr as minimization mode

9) enter the measured data (see previous page) selecting the proper units

10) click on calculate button

P Rel 1.28.54 License 400	0 support ends 20241212 u	user PRODE					×	
Feeds	Streams	Config	Chemica	als	BIPs	N	lodels	
	U21-T				0			
	Save			S	ave / store in Fi	le		
n	Nodel for vapor phase		PRX-NRTL(P-	HV)			~	
M	Nodel for liquid phase		PRX-NRTL(P-I	HV)			\sim	
	Model for solid phase		SPRX-NRTL(P	-HV)			\sim	
	Regress		measured VLI	E-LLE-SLE data	points		~	
	Bips data set to solve		Standard set, more accurate but slow					
	Minimization mode		F = xerr * yerr					
	Calculate				Clear table			
Туре	X1	,	/1	Tempe	rature	Pres	sure	
SLE ~	0.9432		1	266.85	к ~	0	bar.g $$	
SLE V	0.9		1	259.65	к ~	0	bar.g 🗸	
SLE ~	0.87676		1	257.65	к ~	0	bar.g 🗸	
SLE ~	0.80583		1	246.85	к ~	0	bar.g \lor	
SLE ~	0.8		1	242.95	к ~	0	bar.g 🗸	
SLE 🗸	0.728		1	233.45	к ~	0	bar.g 🗸	
SLE 🗸	0.641		1	217.95	к ~	0	bar.g 🗸	
SLE 🗸	0.636		1	214.95	к 🗸	0	bar.g 🗸	
SLE 🗸	0.6		1	208.15	к ~	0	bar.g 🗸	

11) the procedure shows (on second table) the calculated parameters (U12,U21,A12) and (last table, on right of measured values) the calculated values and the relative errors

12) click on button Save to save the values calculated in memory or Save / store in File to save in File.

Now we wish to test the accuracy of calculated BIPs in estimating freezing point depression.

In Prode Properties Editor select stream 9 and define as composition C1=0.0 C2=0.0 H2O=0.636 CH4O=0.364 to test this point in the series (for a mixture with two components the fraction of CH4O = 1.0 - fraction of H2O) Point X(1) Y(1) Temperature (K) Pressure (Bar.g)

0

Point	X(1)	¥(1)	1	emperatu	e (K)		
SLE	0.636	1	2	14.95			
P Rel 1.28.	54 License 4000 suppor	rt ends 20241212 user l	PRODE		×		
Feeds	Streams	Config	Chemicals	BIPs	Models		
	Select feed		9 test freezing poi	nt	~		
	Feed name			test freezing point	t		
	Flow units		Molar flow		~		
	Flow (stream	0	1	kmol/s	~		
	Reaction set	t	Reaction set 1				
	Action		Balance Chemical Equations				
	Balanced chemical e	quation					
	Chemical		ABIETIC ACID		C20H3(🗸		
	Sorting criter	ia	Sort by first name		~		
A	dd component	Remove	component	Clea	ar list		
	Component	React	ion set 1	Molar	fraction		
	METHANE	No	\sim		0		
	ETHANE	No	~		0		
	WATER	No	~	0.	.636		
	METHANOL	No	~	0.	364		
			~		0		

verify that the models for fugacity are based on PRX-NRTL(P-HV)

Predefined packages	1 SRK(VDW)	\sim	SRK(VDW)		Save	
	Vapor		Liquid		Solid	
Fugacity	PRX-NRTL(P-HV)	\sim	PRX-NRTL(P-HV)	\sim	SPRX-NRTL(P-HV)	
Enthalpy	PRX-NRTL(P-HV)	\sim	PRX-NRTL(P-HV)	\sim	REGULAR	
Entropy	PRX-NRTL(P-HV)	\sim	PRX-NRTL(P-HV)	\sim	REGULAR	
Volume	PRX(VDW)	\sim	PRX(VDW)	\sim	REGULAR	
Multipha	se equilibria		Multiphase vapor-lie	quid-sol	id	
Multiphas	e initialization		Standard tests			
Detect	Phase State		From Gibbs / Isothe	rmal Co	mpr. and Liq.Dens.	
Stability analysis, ch	eck stability against feed		Increased thresold (conserva	ative tests)	
Phase diagram, s	pecified fraction lines		End crossing phase	bounda	ry	
Hydrate stru	ctures inclusion		Include normal strue	ctures ge	enerated by formers	
Source	e for BIPs		Prode VLE dataset			
	Save			Save / st	ore in File	

then click on Save button to update the stream 9

In Streams tab select T-P Flash as operation to solve (isothermal flash at specified T and P) then enter the specifications

temperature 214.5 K Pressure 0.0 Bar.g

And click on button Compute to solve the operation

el 1.28.54 Lice	nse 4000 support	ends 20241212 user	PRODE				
Feeds	Streams	Config	Chemicals		BIPs	Mod	els
Operation	n to solve	T-P Flash		\sim	c	ompute	
Fee	d(s)	9		\sim	1 Test Case 1		
Proc	duct	9		~	Connect produ	ct to feed	
Spec	. (IN)	214.5	к	\sim	0	bar.g	
Spec.	(OUT)			×.,			
Select Strea	am to show	9		\sim	Flows (mole)		
Stream O	perating	214.5	K	~	1.013274	bar.a	
Phase		Feed	Liqu	uid		Solid	
Flow (kmol/	s)	1.0252421	1.022	29959		0.002246216	1
Energy (kW	D	83773.156	8366	9.286		103.86961	
Fractions (mo	lar)	1	0.9978091			0.0021909128	
CH4		0	0		0		
C2H6		0		0		0	
H2O		0.636	0.635	20076	5	1	
СН4О		0.364	0.364	79924	4	1.4999801e-0	9
		0		0		0	
		0		0		0	

in the same way it is possible to test the second point (C1=0.0 C2=0.0 H2O=0.9 CH4O=0.1 T=259.65 K P=0.0 Bar.g)

Kel 1.20.54 License	4000 support	enus 20241212 user	PRODE				^
Feeds	Streams	Config	Cher	micals	BIPs	Mod	els
Operation to	solve	T-P Flash		\sim		Compute	
Feed(s)		9		\sim	1 Test Case 1		
Product	t	9		\sim	Connect prod	uct to feed	~
Spec. (IN	4)	259	к	\sim	0	bar.g	~
Spec. (OU	т)		-				
Select Stream	to show	9		\sim	Flows (mole)		~
Stream Oper	ating	259	К	~	1.013274	bar.a	
Phase		Feed		Liquid		Solid	
Flow (kmol/s)	12	1.1907054		1.078184	6	0.11252081	
Energy (kW)		74184.631		68816.17	7	5368.4538	
Fractions (molar)	2	1		0.9055007	'1	0.094499286	5
CH4		0		0		0	
C2H6		0		0		0	
H2O		0.9		0.8895638	36	1	
CH40		0.1		0.1104361	4	1.7233237e-0	9
	15	0		0		0	

Working with archives

Prode Properties stores data in different files

Chemical's data	: chema.dat, chemb.dat
Pseudo-component's data	: pseudo.dat
Binary Interaction parameter's data	: bips.dat
Model's data	: mod.dat
Feeds, Units of measurement, Configuration data	: def.ppp

Prode Properties editor allows to inspect and modify these archives, the different pages include buttons to overwrite these files, the user can modify existing values, add new components etc.

For example, it is possible to edit / modify / add new Feeds, Units of Measurement, Configuration parameters and then store the new values in a file so that the new information will not be lost when the user ends the program :

P Rel 1.28.	P Rel 1.28.3 License 2490 support ends 20221230 user acme company					
Feeds	Streams	Config	Chemicals E	BIPs Models		
	• • • •					
	Calorific Value		kJ/kg			
Ca	alorific Value (mol	ar)	kJ/kmol			
I	Enthalpy (Streams	5)	kW			
	Entropy (Streams)	kJ/(K*s)			
	Heat Capacity		kJ/(kg*K)			
н	eat Capacity (mol	ar)	kJ/(kmol*K)			
Base	value for enthalp	y calc.	Specified value and ter	mperature		
Base t	emperature for er	nthalpy	1	к		
Ba	se value for entha	alpy	5000	kJ/kg		
Base	value for entropy	/ calc.	Specified value and ter	mperature		
Base	temperature for e	ntropy	1	к		
Ba	ase value for entro	ру	50	kJ/(kg*K)		
Convergen	ce tolerance on sp	pecifications	1	e-09		
Max allowe	ed time for solving	g operations	60	5		
	Flow units		Molar flows			
Minimu	ım Density for liqu	uid state	200	kg/m3		
Set r	new configuration v	alues	Store	e in File		

Introducing Prode Properties library methods

Prode Properties library includes a range of methods to deal with problems in chemical engineering and to achieve tight control over the calculations .

A non-inclusive list would include

- Thermodynamic calcs (flash operations, enthalpy, entropy, volume, energy, unit operations)
- Streams data access and calcs (set and retrieve operating conditions, critical and transport properties calcs)
- Chemicals library access (retrieve data from chemicals file)
- · Error messages (management of errors messages)

Methods for thermodynamic calc's

Prode Properties includes a complete set of methods for solving all the standard flash operations with specified final temperature or pressure and entropy or enthalpy or volume or energy basis, phase fraction with temperature or pressure basis plus mixers, dividers, gas, liquid phase separation operations etc.

integer result = setOp(integer stream, double t, double p) sname xftp

Given a stream, operating pressure and temperature, performs an isothermal flash and sets operating conditions.

integer result = setSOp(integer stream) sname xfstp Given a stream performs an isothermal flash at (user defined) standard conditions.

double t = PfPF(integer stream, double p, double pf, int state, int n)

Sname xfpfp

Given a stream, the pressure , phase fraction (range 0-1), state (gas, liquid, solid) and position n calculates and returns the nth (n : 1-5) equilibrium temperature along the specified phase fraction line

double p = PfTF(integer stream, double t, double pf, int state, int n) sname xfpft

Given a stream, the temperature , phase fraction (range 0-1), state (gas, liquid, solid) and position n calculates and returns the nth (n : 1-5) equilibrium pressure along the specified phase fraction line

double t = LfPF(integer stream, double p, double lf)

sname xflfp

Given a stream, the pressure and Liquid fraction (range 0-1) calculates and returns the first equilibrium temperature along the specified phase fraction line

double p = LfTF(integer stream, double t, double lf)

sname xflft

Given a stream, the temperature and Liquid fraction (range 0-1) calculates and returns the first equilibrium pressure along the specified phase fraction line

double t = BPF(integer stream, double p) sname xfbp Given a stream and pressure calculates and returns bubble point temperature

double t = DPF(integer stream, double p) sname xfdp Given a stream and pressure calculates and returns dew point temperature

double t = HPF(integer stream, double p, double h, double et) sname xfhp

Given a stream, final pressure, the required (final) enthalpy (see the method StrH() for the definition) and a estimated value for final temperature (or 0 for automatic estimate), method solves the flash operation (enthalpy basis) and returns final temperature

double p = HTF(integer stream, double t, double h, double ep) sname xfht

Given a stream, final temperature, the required (final) enthalpy (see the method StrH() for the definition) and a estimated value for final pressure (or 0 for automatic estimate), method solves the flash operation (enthalpy basis) and returns final pressure

double t = SPF(integer stream, double p, double s, double et) sname xfsp Given a stream, final pressure, the required (final) entropy (see the method StrS() for the definition) and a estimated value for final temperature (or 0 for automatic estimate), method solves the flash operation (entropy basis) and returns final temperature. double p = STF(integer stream, double t, double s, double ep) sname xfst Given a stream, final temperature, the required (final) entropy (see the method StrS() for the definition) and a estimated value for final pressure (or 0 for automatic estimate), method solves the flash operation (entropy basis) and returns final pressure. double t = VPF(integer stream, double p, double v, double et) sname xfvp Given a stream, final pressure, the required specific volume (see the method StrV() for the definition) and a estimated value for final temperature (or 0 for automatic estimate), method solves the flash operation (volume basis) and returns final temperature. double p = VTF(integer stream, double t, double v, double ep) sname xfvt Given a stream, final temperature, the required specific volume (see the method StrV() for the definition) and a estimated value for final pressure (or 0 for automatic estimate), method solves the flash operation (volume basis) and returns final pressure. integer result = HVF(integer stream, double h, double v, double et, double ep) sname xfhv Given a stream, the required (final) enthalpy (see the method StrH() for the definition) the required (final) specific volume (see the method StrV() for the definition) and estimated values for final temperature and pressure (or 0 for automatic estimate), method solves the flash operation integer result = SVF(integer stream, double s, double v, double et, double ep) sname xfsv Given a stream, the required (final) entropy (see the method StrS() for the definition) the required specific volume (see the method StrV() for the definition) and estimated values for final temperature and pressure (or 0 for automatic estimate), method solves the flash operation integer result = HSF(integer stream, double h, double s, double et, double ep) sname xfhs Given a stream, the required (final) enthalpy (see the method StrH() for the definition) the required (final) entropy (see the method StrS() for the definition) and estimated values for final temperature and pressure (or 0 for automatic estimate), method solves the flash operation double t = EPF(integer stream, double p, double E, double aout, double et) sname xfep Given a stream, final pressure, outlet area, the term E (equal to Hin + 1/2Vin^2) and a estimated value for final temperaure (or 0 for automatic estimate) method solves the constant energy flash and returns final temperature, method solves Hin + 1/2Vin² = Ho + 1/2Vo² and it permits to model adiabatic, irreversible expansions when the contribute of kinetic energy cannot be neglected. integer result = MixF(integer stream1, integer stream2, double et) sname xmix Given two streams, stream1 and stream2 and a estimated value for final temperature (or 0 for automatic estimate) method solves a mixer operation and returns the result on stream1, the feed streams are adiabatically flashed to the lowest inlet stream pressure integer result = Divi (integer stream1, integer stream2, double wdiv)

sname xdivi

Given two streams (stream1 and stream2) and a flowrate fraction (0-1) performs a divider operation so that stream 1 is shifted into two streams (stream1, stream2) of the same composition, temperature and pressure, flowrate fractions are subdivided as specified by wdiv (stream2 = wdiv, stream1 = 1- wdiv)

integer result = psep(integer stream1, integer stream2, integer phase) sname xpsep

Given a stream (stream1) performs an isothermal flash to simulate a phase type (vapor,liquid,solid) separator and returns the result as stream2.

integer res = StrCopy(integer stream1, integer stream2)

sname xscopy

Given two streams (stream1 and stream2) copies the stream 2 into stream 1, the method copies all valid data including operating data if available.

Methods for stream's data access

Prode Properties includes a set of functions for accessing stream parameters and calculating transport properties.

integer res = isSDef(integer stream) sname xsdef given a stream returns TRUE (value = 1) if stream has been defined, otherwise returns FALSE (0)

double t = getT(integer stream) sname xst given a stream returns stream's operating temperature

double p = getP(integer stream) sname xsp given a stream returns stream's operating pressure

integer nr = getPNr() sname xpnr returns the maximum number of phases that procedure can detect

integer type = StrPt(integer stream, int phase) sname xspt given a stream and position in range 1- getPNr() returns the phase type (vapor,liquid,solid)

char *description = StrPts(integer stream, int phase) sname xspts given a stream and position in range 1- getPNr() returns a ANSI C string with the description (vapor, liquid, solid...)

int description MStrPts(integer stream, int phase, char *s, integer slm) given a stream and position in range 1- getPNr() fills string s with the description (vapor, liquid, solid...) (eventually truncated to slm maximum lenght), this is the Microsoft Excel specific method

double If = StrLf(integer stream) sname xslf given a stream returns the total liquid fraction (molar basis) in stream

double pf = StrPf(integer stream, integer phase) sname xspf given a stream and phase position in range 1- getPNr() returns the phase fraction

double w = getW(integer stream, integer phase, integer pos.) given a stream, the phase position and component's position (in component's list) returns the component molar fraction in that phase

double Zi= getZ(integer stream, integer pos.) sname xsz given a stream and component's position (in component's list) returns the comp's Z (weight percentage, molar basis) integer res = putZ(integer stream, integer pos., double Zi) sname xsetsz given a stream, comp's position and Z , sets the comp's pos. in Z vector (composition, molar basis) for that stream

integer nr = getCNr(integer stream) sname xscnr given a stream returns the number of components defined in that stream

integer nr = getMCNr() sname xsmcnr returns the maximum number of components in a stream

double zv = StrZv(integer stream) sname xszv given a stream returns the relevant compressibility factor (gas phase)

double mw = StrMw(integer stream) sname xsmw given a stream returns the averaged molecular weight (all phases)

double v = StrV(integer stream) sname xsv given a stream returns the specific volume as sum of specific volumes of all phases

double mw = StrGMw(integer stream) sname xsgmw given a stream returns the averaged molecular weight (gas phase)

double mw = StrLMw(integer stream) sname xslmw given a stream returns the averaged molecular weight (liquid phase)

double h = StrH(integer stream) sname xsh given a stream returns the total (stream) enthalpy (gas + liquid + solid phases)

double h = StrGH(integer stream) sname xsgh given a stream returns the total (stream) enthalpy (gas phase)

double h = StrSGH(integer stream) sname xssgh given a stream returns the specific (unit weight) enthalpy (gas phase)

double h = StrLH(integer stream) sname xslh given a stream returns the total (stream) enthalpy (liquid phase)

double h = StrSLH(integer stream) sname xsslh given a stream returns the specific (unit weight) enthalpy (liquid phase)

double h = StrSH(integer stream) sname xssh given a stream returns the total (stream) enthalpy (solid phase)

double h = StrSSH(integer stream) sname xsssh given a stream returns the specific (unit weight) enthalpy (solid phase) double cp = StrGICp(integer stream) sname xsgicp given a stream returns the ideal gas heat capacity

double cp = StrGCp(integer stream) sname xsgcp given a stream returns the specific heat capacity (constant pressure, gas phase)

double cv = StrGCv(integer stream) sname xsgcv given a stream returns the specific heat capacity (constant volume, gas phase)

double cp = StrLCp(integer stream) sname xslcp given a stream returns the specific heat capacity (constant pressure, liquid phase)

double cv = StrLCv(integer stream) sname xslcv given a stream returns the specific heat capacity (constant volume, liquid phase)

double cp = StrSCp(integer stream) sname xsgcp given a stream returns the specific heat capacity (constant pressure, solid phase)

double ss = StrMSS(integer stream) sname xsmss given a stream returns the speed of sound (gas, liquid) as calculated with HEM model for mixed phases

double ss = StrGSS(integer stream) sname xsgss given a stream returns the speed of sound in gas phase

double ss = StrLSS(integer stream) sname xlmss given a stream returns the speed of sound in liquid phase

double jt = StrGJT(integer stream) sname xsgjt given a stream returns the Joule Thomson coefficient in gas phase

double jt = StrLJT(integer stream) sname xsljt given a stream returns the Joule Thomson coefficient in liquid phase

double ic = StrGIC(integer stream) sname xsgic given a stream returns the isothermal compressibility coefficient - (1 / V) * dV / dP in gas phase

double ic = StrLIC(integer stream) sname xslic given a stream returns the isothermal compressibility coefficient - (1 / V) * dV / dP in liquid phase

double v = StrGVE(integer stream) sname xsgve given a stream returns the volumetric expansivity coefficient - (1 / V) * dV / dT in gas phase

double ic = StrLVE(integer stream) sname xslve given a stream returns the volumetric expansivity coefficient - (1 / V) * dV / dT in liquid phase double s = StrGS(integer stream) sname xsgs given a stream returns the total (stream) entropy (gas phase)

double s = StrSGS(integer stream) sname xssgs given a stream returns the specific (unit weight) entropy (gas phase)

double s = StrLS(integer stream) sname xsls given a stream returns the total (stream) entropy (liquid phase)

double s = StrSS(integer stream) sname xsss given a stream returns the total (stream) entropy (solid phase)

double s = StrSLS(integer stream) sname xssls given a stream returns the specific (unit weight) entropy (liquid phase)

double s = StrSSS(integer stream) sname xssss given a stream returns the specific (unit weight) entropy (solid phase)

double s = StrS(integer stream) sname xss given a stream returns the total (stream) entropy (gas + liquid + solid phases)

integer res = setWm(integer stream, double W) sname xsetswm given a stream and flow (mass basis), sets the flow

double w = getWm(integer stream) sname xswm given a stream returns the flow specified for that stream.

double hc = StrHC(integer stream) sname xshc given a stream returns the calculated net heat of combustion (gas phase).

double fl = StrFML(integer stream) sname xsfml given a stream returns the calculated flammability lean limit (gas phase).

double fl = StrFMH(integer stream) sname xsfmh given a stream returns the calculated flammability rich limit (gas phase).

double d = StrLD(integer stream) sname xsld given a stream returns the calculated liquid density (at operating conditions)

double d = StrGD(integer stream) sname xsgd given a stream returns the calculated gas density (at operating conditions)

double tc = StrLC(integer stream) sname xsgd given a stream returns the calculated liquid thermal conductivity (at operating conditions) double tc = StrGC(integer stream) sname xsgc given a stream returns the calculated gas thermal conductivity (at operating conditions)

double v = StrLV(integer stream) sname xslv given a stream returns the calculated liquid viscosity (at operating conditions)

double v = StrGV(stream) sname xsgv given a stream returns thecalculated gas viscosity (at operating conditions).

double st = StrST(integer stream) sname xsst given a stream returns the calculated surface tension (at operating conditions).

Integer cpnr = StrCPnr(integer stream) sname xscpnr given a stream returns the number of critical points detected and calculated, to get a critical point use the methods StrPc() And StrTc() setting value of pos in the range 1-cpnr

double p = StrPc(integer stream, Integer pos) sname xspc given a stream and the critical point position in the list (see method StrCPnr()) returns the critical pressure

double t = StrTc(integer stream, Integer pos) sname xstc given a stream and the critical point position in the list (see method StrCPnr()) returns the critical temperature.

double p= StrCBp(integer stream) sname xscbp given a stream returns the cricodenBar pressure.

double t= StrCBt(integer stream) sname xscbt given a stream returns the cricodenBar temperature.

double p= StrCTp(integer stream) sname xsctp given a stream returns the cricodenTherm pressure.

double t= StrCTt(integer stream) sname xsctt given a stream returns the cricodenTherm temperature.

double ac = StrAc(integer stream) sname xsac given a stream returns the acentric factor (mole fraction average).

double p= StrRVP(integer stream, integer mode) sname xsrvp given a stream returns the Reid vapor pressure mode = 1 simulation of D6377 procedure (liquid not saturated with air) mode = 2 simulation of D323 procedure (liquid saturated with air)

double fp = StrFLP(integer stream)

sname xsflp

given a stream returns the Flash point (for pure fluids the method returns the value stored in databank while for mixtures the flash point is calculated by a iterative procedure where VLE is solved according the selected models for stream)

Methods to work with packages

Each package stores a set of models for fugacity, enthalpy, entropy, volume and the different states (vapor, liquid, solid, hydrate), the library includes methods to define, store and edit packages

Integer nr = getPKnr() xname xpknr return the max number of packages

Integer nr = getPKdnr() xname = xpknr return the number of packages with valid data

char *str= getPKN(int pkg) xname = xpkn given the package this method returns the name

integer res = putPKN(int pkg, char* name) xname = xsetpkn given the package and the name this method sets the name

nteger putPKM(int pkg, int prop, int state, int model) xname = xsetpkm given the package, property, state and model the method sets the model and return true

integer getPKM(int pkg, int prop, int state) xname = xpkm given the package, property and state the method returns the model

integer res = putPKS(int pkg, int option, int value) xname = xsetpks given the package, property, option and value the method sets the option and return true

integer getPKS(int pt, int option) xname = xpks given the package and option the method returns the value

Methods to work with streams

Each stream stores a list of components and molar fractions, the associated models etc. the library includes methods to define, store and edit streams

to define a stream :

- call initS()
- for each component in the list set the component's code with putCC() set the component's mole fraction with putZ()
- call setS() to define the stream
- call setW() to define the flow
- utilize the methods described in paragraph "Methods to define thermodynamic models" to define the models
- call loadSB() to load the BIPs from database or define specific BIPs with methods PutCi(), PutCj(), PutMB(), PutBIP()

or, to simply change the component's fractions :

- for each component in the list
 - set the new component's mole fraction with putZ()
- call setS() to define the stream

List of methods exported

integer res = initS (integer stream) sname xinits given a stream initializes all data, call this method before to create a new list of components.

integer res = putCC (integer stream, integer pos, integer compcode) sname xsetscc given a stream, component's position (in component's list) and component code sets the code in component's list.

integer res = putZ(integer stream, integer pos., double Zi) sname xsetsz given a stream, comp's position and Z , sets the comp's pos. in Z vector (composition, molar basis) for that stream

integer res = setS(integer stream) sname xsets given a stream performs a sequence of validating operations on data. This method must be called after to have restored stream's data from archives (files etc.)Methods to define a initial condition for a stream

nteger res = loadSB(integer stream, integer btype) sname xloadsb

given a stream loads all BIP available in database. This method must be called after the stream has been defined since it requires the list of components. Codes for btype are 0 for VLE, 1 for LLE, 2 for SLE, 3 for Hydrates

double Zi= getZ(integer stream, integer pos.) sname xsz given a stream and component's position (in component's list) returns the comp's Z (molar fraction)

integer cc = getCC(integer stream, integer pos) sname xscc given a stream and component's position (in component's list) returns the component code (a integer that identifies the component in chemical's file).

integer nr = getMBPNr() sname xsmbnr returns the maximum number of (interaction coefficients) binary pairs in a stream

int ci = getCi(integer stream, integer pos)
sname xsci
given a stream and position (in interaction's coeff. list) returns the first component reference (a integer that identifies the
component in component's list)

integer res = PutCi (integer stream, integer pos, integer ci) sname xsetsci

given a stream, position (in interaction coefficients list) and first component reference sets the component's reference in interaction coefficient's list.

int cj = getCj(integer stream, integer pos)

sname xscj

given a stream and position (in interaction's coeff. list) returns the second component reference (an integer that identifies the component in component's list)

integer res = PutCj (integer stream, integer pos, integer cj) sname xsetscj

given a stream, position (in interaction coefficients list) and second component reference sets the component's reference in interaction coefficient's list

int model = getMB(integer stream, integer pos) sname xsmb given a stream and position (in interaction's coeff. list) returns the related model (an integer that identifies the model).

integer res = PutMB(integer stream, integer pos, integer model) sname xsetsmb

given a stream, position (in interaction coefficients list) and a model identifier sets the model in interaction coefficient's list.

double BIP = getBIP(integer stream, integer pos, integer id) sname xsbip given a stream, position (in binary coeff. list) and BIP identifier (0-max nr. of BIPs for that model) returns BIP.

integer res = PutBIP(integer stream, integer pos, integer id. double Kji)

sname xsetsbip

given a stream, position (in binary coeff. list) BIP identifier (0-max nr. of BIPs for that model) and value stores BIP in that position of the list.

Methods to define stream's operating conditions

Prode Properties includes a set of functions to define phase fractions, the different phase's compositions etc. in a operating stream, these can be utilized, for example, to enter data calculated with another software

- call rstValidSop()
- for each phase
- for each component define fraction with putW()
- define phase fraction with putPF()
- define phase type with putPT()
- set phase as valid , setValidPhase()
- define temperature with putT()
- define pressure with putP()
- set conditions as valid with setValidSop()

List of methods exported

integer result = rstValidSop(integer stream) Sname xrstvop Given a stream clears the compostions of different phases at operating conditions integer result = setValidSop(integer stream) sname xsetvop

Given a stream sets the compostions of different phases at operating conditions.as valid.

integer result = setValidPhase(integer stream, integer phase) sname xsetvphase Given a stream and phase sets the phase compostion.as valid.

integer result = putW(integer stream, integer phase, int compnr, double w) sname xsetw Given a stream, phase, component number and component's molar fraction in that phase stores the value

integer result = putPF(integer stream, integer phase, double fraction) sname xsetpf Given a stream, phase and phase fraction stores the phase .fraction value

integer result = putPT(integer stream, integer phase, int type) sname xsetpt Given a stream, phase and phase type (vapor,liquid,solid) stores the phase type

nteger result = putT(integer stream, double t) sname xsetst Given a stream and operating temperature stores the value

nteger result = putP(integer stream, double p) sname xsetsp Given a stream and operating pressure stores the value

Methods for solving staged columns

Properties includes a procedure for solving staged columns (versions for continuous and batch distillation), the column is modeled with stgnr equilibrium stages, column may include a condenser and a reboiler, stage numbering is bottom up, the bottom stage (reboiler, if specified) is number one and the top stage (condenser, if specified) is number stgnr. There may be one or more feeds, a feed is modeled by entering liquid on the specified stage and vapor portion to the stage above (with exception of top stage).

There may be one or more side streams

Heat added / removed on each stage can be specified

Efficiency parameter on each stage can be specified

integer res = DCOL(int csep, int stgnr, int init, double *stgt,double *stgp,double *stgef,double *stgdH, int prod_h, int btm_h, int fnr,int *fstr,int *fpos,int snr,int *sstr,int *spos,int *sft, double *sflow,int vnr,double *vrv,int *vtype,int *ptype,int *piv,double *prv, double*flows)

Parameters :

csep	(int)	column type : 1 VLE , 2 VLLE , 3 LLE (some features available in extended versions)
stgnr	(int)	number of stages
init	(int)	0 for automatic initialization, 1 temperatures and flows are defined by user
stgt	(double*)	vector (stgnr) with stage temperatures
stgp	(double*)	vector (stgnr) with specified stage pressures
stgef	(double*)	vector (stgnr) with specified stage efficiency, permitted range 0,1-1
stgdH	(double*)	vector (stgnr) with specified dH (heat added, removed)
prod_h	(int)	stream for top product/distillate
btm_h	(int)	stream for bottom product
fnr	(int)	number of feeds
fstr	(int*)	vector (fnr) with the feeding streams
fpos	(int*)	vector (fnr) with feeds positions 1-stgnr
snr	(int)	number of side streams
sstr	(int*)	vector (snr) with the list of side streams
spos	(int*)	vector (snr) with side streams positions (1-stgnr)
sft	(int*)	vector (snr) with specified flow type (GAS_PHASE, LIQ_PHASE, see Codes used in Prode library)
sflow	(double*)	vector (snr) with the specified (on each side stream) side product to feed flow ratio
vnr	(int)	number of variables to solve
vtype	(int*)	vector (vnr) with type of variable (seebelow)
vrv	(double*)	vector (vnr) with calculated values for variable
ptype	(int*)	vector (pnr) with type of specification (see below)
piv	(int*)	vector (pnr) with integer values as the position of components in the list
prv	(double*)	vector (pnr) with values of the specifications to solve
flows	(double*)	vector with calculated values for vapor/liquid flows in all stages, dimension nrphases*nrc*stgnr when a condenser is present the reflux is the liquid flow on top stage

Notes :

When passing / returning paramenters the first element in vectors is the element 0

Main variables (1-vnr) are (when specified) reboiler and condenser (partial or total), each variable (of type defined in vtype) requires a suitable specification (in ptype, piv, prv), usually for reboiler the specification is the product to feed ratio and for a condenser the reflux ratio, but specifications based on component's fractions on top and bottom products are permitted, in these cases specify in piv the position of selected component in the list and in prv the value of the fraction required Secondary variables are side streams (1-snr), each side stream (defined in sstr, spos) requires (in sflow) a specification for the side product to (total) feed flow ratio.

The column is modeled with thermodynamics and options defined for the first feed in the list. Initialization

in most cases the procedure doesn't require to initialize values, when required set the variable init to 1 and define the initial values in vectors stgt and flows, note that in a sequence of similar operations (for example when controlling the operating point of a column) it may result useful to reintroduce the calculated values as starting point for the new calculus

Methods for solving staged columns (continuation)

Codes for variables

reboiler	1
total condenser	2
partial condenser	3

Codes for specifications

reflux ratio	1
product to feed ratio (molar fract.)	2
bottom to feed ratio (molar fract.)	3
component (molar fract.) in top product	4
component (molar fract.) in bottom product	5
component recovery in top product	6
component recovery in bottom product	7

Example

Column with 8 stages, 1 feed (stage 4), pressure reboiler 12.5 Bar, pressure top 12 Bar, stage efficiency 1.0, dH = 0.0 variables : reboiler and total condenser

specifications :	component 2 fraction in t	top product and bottom product to to feed ratio
parameter	value	comment
csep	1	VLE column
stgnr	8	number of stages
init	0	automatic initialization
stgp[0]	12.5	pressure on stage 1
		specify pressure for all stages
stgp[7]	12	pressure on stage 8
stgef[0]	1	efficiency on stage 1
		specify efficiency for all stages
stgef[7]	1	efficiency on stage 8
stgdH[0]	0	heat added, removed on stage 1
		specify heat added / removed for all stages
stgdH[7]	0	heat added, removed on stage 8
prod_h	1	product stream (Prode Properties stream 1)
btm_h	2	bottom stream (Prode Properties stream 2)
fnr	1	feeds number
fstr	3	feed stream (Prode Properties stream 3)
fpos	4	feed position
vnr	2	number of variables
vtype[0]	1	first variable, reboiler
vtype[1]	2	second variable, total condenser
ptype[0]	4	first specification, molar fraction in top product
piv[0]	2	first specification, second component (2) in the list
prv[0]	0.96	first specification, required fraction
ptype[1]	3	specification, bottom to feed ratio
piv[1]	0	not required
prv[1]	0.4	second specification, required value (bottom to feed ratio = 0.4)

Methods for solving reactors

simulation of reactors with standard procedures

int res = REACT(int streamIn, streamOut, int model, int NrReactions, double **Conv, double Pout, double dHeat)

Parameters :		
streamIn	(int)	inlet stream
streamOut	(int)	outlet stream
model	(int)	model for reactor (see below)
NrReactions	(int)	number of reactions
Conv	(double**)	matrix (NrComponents, NrReactions) to specify reactions
Pout	(double)	output pressure
dHeat	(double)	heat added, removed
Codes for mod	lels	

Gibbs1Equilibrium Reactor2

(**) additional models available from Prode

Methods for solving fluid flow problems

simulation of single phase, two-phases, multiphase flow on circular pipes

int res = PIPE(int stream, int model, double diam, double rough, double length, double dHeight, double dHeat)

Parameters :	
stream (int)	inlet stream
model (int)	model for fluid flow and phase equilibria (see the codes below)
diam (double)	pipe internal diameter
rough (double)	parameter defining relative pipe roughness
length (double)	lenght of this segment
dHeight (double)	height difference (inlet, outlet)
dHeat (double)	heat added, removed
codes for models	

(**) additional models available from Prode

Methods for Hydrates phase equilibria

methods for calculating hydrate formation pressure (or temperature)

1

double p = HPFORM(int stream, double t, int method)

double t = HTFORM(int stream, double p, int method)

Parameters :

Beggs & Brill

stream	(int)	inlet stream
t, p	(double)	operating temperature (or operating pressure)
method	(int)	1 = include SI , SII , SH
		2 = SI
		3 = SII

Methods for solving Polytropic operations

Polytropic stage (compression and expansion) rigorous models for compressors and expanders including phase equilibria

double val = PSPF(int stream, double pout, int model, double param) sname xpolp

Parameters :	
stream (int)	inlet stream
pout (double)	outlet pressure
model (int)	model, see below codes 1-4
param (double)	for model 1 and 3 specified polytropic efficiency (range 0-1)
	for model 2 and 4 (measured) outlet temperature

the procedure can model compression and expansion units such as centrifugal compressors, expansion turbines etc. including phase equilibria

the procedure returns -calculated temperature options 1,3 -calculated efficiency options 2,4

- 1 given initial condition, pout and polytropic efficiency calculates outlet condition, R.A. Huntington "Evaluation of Polytropic calculation Methods for Turbomachinery Performance", method applicable to gas phase only
- 2 given initial condition, pout and tout calculates polytropic efficiency, R.A. Huntington "Evaluation of Polytropic calculation Methods for Turbomachinery Performance", method applicable to gas phase only
- 3 given initial condition, pout and polytropic efficiency calculates outlet condition, R.Paron "Polytropic solution with phase equilibria" method applicable to gas and mixed (gas + liquid) phases
- 4 given initial condition, pout and tout calculates polytropic efficiency, R.Paron "Polytropic solution with phase equilibria" method applicable to gas and mixed (gas + liquid) phases

(**) additional models available from Prode

Methods to design / rate orifices and relief valves

This unit models a relief valve (vapor and liquid phases) at specified operating conditions and returns the calculated area

double area = ISPF(int stream, double pout, int model, double *param)

Parameters :	
stream (int)	inlet stream
pout (double)	outlet pressure
model (int)	model, see below codes 1-4
param(double)	correction parameter, see below the range of recommended values
models available (**)	
1	HEM Homogeneous Equilibrium (Solution of Mass Flux integral)
2	HNE Homogeneous Non-equilibrium (HEM with Boling Delay and Gas-Liquid Slip Contributes)
3	HNE-DS, Homogeneous Non-equilibrium
4	NHNE Non-homogeneous Non-equilibrium
recommended range of	values for correction parameter
HEM	not required
HNE	0.7-0.8 for safety valves
HNE-DS	see the paper
NHNE	0.7-0.8 for safety valves

(**) additional models available from Prode

Methods for calculating equilibrium lines in phase diagrams

Prode Properties includes methods for calculating different types of phase diagrams vapor-liquid vapor-liquid-liquid vapor-liquid-solid (**) (**) feature available in custom versions

typical application

- define the stream, set the required phase equilibria (vapor-liquid, vapor-liquid, vapor-liquid, vapor-liquid-solid)
- call PELnr() to calculate the phase diagram and obtain the number of lines available
- on each line call PELP(), PELT(), PELine() to obtain the data for the different lines
- if required call PFLine() to calculate a line with specified phase fraction ad state

integer Inr = PELnr(integer stream) sname xpelnr Given a stream calculates the phase diagram and returns the number of equilibrium lines available

integer Inr = PELT(integer stream, integer line) sname xpelt Given a stream and the line, returns the line type (see below) 1 = bubble line

2 = dew line

3 = three phase line

integer Inr = PELP(integer stream, integer line)

sname xpelp

Given a stream and the line, returns the line property (see below)

1 = vapor-liquid

- 2 = vapor-liquid-liquid
- 3 = vapor-solid
- 4 = liquid-solid

integer nrpt =PELine(integer stream, integer line, double *P, double *T, int maxpt)

sname xpel

Given a stream, the line and two arrays (0 -maxpt elements) the procedure returns nrpt < maxpt equilibrium points in specified line

integer nrpt =PVLine(integer stream, integer line, double *P, double *T, double *H, double *S,double *V,int maxpt) sname xpeel

Given a stream, the line and five arrays (0 -maxpt elements) the procedure returns nrpt < maxpt equilibrium points in specified line, in additions to t,p values this method returns enthalpy, entropy and volume values calculated at equilibrium points

integer nrpt =PFLine(integer stream,int line, double pf, double *P, double *T, int maxpt) sname xpepfl

Given a stream, the line, a specified phase fraction and two arrays (0-maxpt elements) the procedure returns nrpt < maxpt equilibrium points in specified phase fraction line

Methods for direct access to properties (F,H,S,V) and derivatives (T,P,W)

Prode Properties includes methods for fast calculations of thermodynamic properties, you can define up to 5 independent processes with method DPinit(), these processes run independently permitting fast executions.

Application example : Process = 1: // range 1-5 Stream=5: // make sure stream 5 has been defined before to call Dpinit() DPinit(process,stream); StrHv(process,0,t,p,X,&HL); StrHv(process,1,t,p,Y,&HV); integer res = DPinit(integer process, integer stream) sname xspi Given a process (code 1-5) and a stream the method loads all data integer res = StrFv(integer process,integer state,double t,double p, double *w,double *fg) sname xsfv Given a predefined stream the required state and operating conditions returns the vector of fugacities(Pa) integer res = StrFvd(integer process, integer state, double t, double p, double *w, double *fg, double *dfgt, double *dfgp, double **dfgw) sname xsfvd Given a predefined stream the required state and operating conditions returns the vector of fugacities (Pa) and related derivatives vs. temperature (K), pressure (Pa), composition (note : derivatives vs. composition as matrix [n][m]) integer res = StrFvdv(integer process, integer state, double t, double p, double *w, double *fg, double *dfgt, double *dfgp, double *dfgw) sname xsfvdv Given a predefined stream the required state and operating conditions returns the vector of fugacities (Pa) and related derivatives vs. temperature (K), pressure (Pa), composition (note : derivatives vs. composition as vector [n*m]) integer res = StrHv(integer process, integer state,double t,double p, double *w,double *H) sname xshv Given a predefined stream the required state and operating conditions returns the molar enthalpy (Ki/ Kmol) integer res = StrHvd(integer process, integer state, double t, double p, double *w, double *H, double *dHt, double *dHp, double *dHw) sname xshvd Given a predefined stream the required state and operating conditions returns the molar enthalpy (Kj/ Kmol) and related derivatives vs. temperature, pressure, composition integer res = StrSv(integer process, integer state, double t, double p, double *w, double *S) sname xssv Given a predefined stream the required state and operating conditions returns the molar entropy (Kj/ Kmol-K) integer res = StrSvd(integer process, integer state, double t, double p, double *w, double *S, double *dSt, double *dSp, double *dSw) sname xssvd Given a predefined stream the required state and operating conditions returns the molar entropy (Ki/ Kmol-K) and related derivatives vs. temperature, pressure, composition integer res = StrVv(integer process, integer state, double t, double p, double *w, double *V) sname xsvv Given a predefined stream, the required state and operating conditions returns the molar volume (M3/Kmol) integer res = StrVvd(integer process, integer state, double t, double p, double *w, double *V, double *dVt, double *dVp, double *dVw) sname xsvvd Given a predefined stream the required state and operating conditions returns the molar volume (M3/ Kmol) and related derivatives vs. temperature, pressure, composition

Methods for stream's data access

Extended methods to obtain properties

These methods are equivalent to standard methods but they add the operating conditions at which the required property must be evaluated. This may result useful in many cases, for example when utilizing Prode Properties methods as macros from Excel cells.

double mw = EStrGMw(integer stream, double t, double p) sname xstpmw given the stream, pressure and temperature performs an isothermal flash and returns the molecular weight for gas phase

double mw = EStrLMw(integer stream, double t, double p) sname xstplmw given the stream, pressure and temperature performs an isothermal flash and returns the molecular weight for liquid phase

double If = EStrLf(integer stream, double t, double p) sname xstplf given the stream, pressure and temperature performs an isothermal flash and returns liquid fraction (molar basis) in stream

double pf = EStrPf(integer stream, integer state, double t, double p) sname xstppf given a stream , state (gas, liquid, solid) pressure and temperature performs an isothermal flash and returns the phase

fraction (molar basis) in specified state

double zv = EStrZv(integer stream, double t, double p) sname xstpzv

given the stream, pressure and temperature performs an isothermal flash and returns the relevant compressibility factor (gas phase)

double h = EStrH(integer stream, double t, double p) sname xstph given the stream, pressure and temperature performs an isothermal flash and returns the enthalpy (gas + liquid phase)

double v = EStrV(integer stream, double t, double p) sname xstpv given a stream, pressure and temperature performs an isothermal flash and returns the specific volume as sum of specific volumes of all phases

double cp = EStrGCp(integer stream, double t, double p) sname xstpgcp given the stream, pressure and temperature performs an isothermal flash and returns the specific heat capacity (constant pressure, gas phase)

double cv = EStrGCv(integer stream, double t, double p)
sname xstpgcv

given the stream, pressure and temperature performs an isothermal flash and returns the specific heat capacity (constant volume, gas phase)

double cp = EStrLCp(integer stream, double t, double p)

sname xstplcp

given the stream, pressure and temperature performs an isothermal flash and returns the specific heat capacity (constant pressure, liquid phase)

double cv = EStrLCv(integer stream, double t, double p) sname xstplcv

given the stream, pressure and temperature performs an isothermal flash and returns the specific heat capacity (constant volume, liquid phase)

double c = EStrGIC(integer stream, double t, double p) sname xstpgic given the stream, pressure and temperature performs an isothermal flash and returns the isothermal compressibility in gas phase double c = EStrLIC(integer stream, double t, double p) sname xstplic given the stream, pressure and temperature performs an isothermal flash and returns the the isothermal compressibility in liquid phase double ss = StrMSS(integer stream, double t, double p) sname xstpmss given the stream pressure and temperature performs an isothermal flash and returns returns the speed of sound (gas, liquid) as calculated with HEM model for mixed phases double ss = EStrGSS(integer stream, double t, double p) sname xstpgss given the stream, pressure and temperature performs an isothermal flash and returns the speed of sound in gas phase double ss = EStrLSS(integer stream, double t, double p) sname xstplss given the stream, pressure and temperature performs an isothermal flash and returns the speed of sound in liquid phase double jt = EStrGJT(integer stream, double t, double p) sname xstpgit given the stream, pressure and temperature performs an isothermal flash and returns the Joule Thomson coefficient for gas phase double it = EStrLJT(integer stream, double t, double p) sname xstpljt given the stream, pressure and temperature performs an isothermal flash and returns the Joule Thomson coefficient for liquid phase double ic = EStrGIC(integer stream double t, double p) sname xstpgic given the stream, pressure and temperature performs an isothermal flash and returns the isothermal compressibility coefficient (1 / V) * dV / dP in gas phase double ic = EStrLIC(integer stream double t, double p) sname xstplic given the stream, pressure and temperature performs an isothermal flash and returns the isothermal compressibility coefficient (1 / V) * dV / dP in liquid phase double v = EStrGVE(integer stream double t, double p) sname xstpgve given the stream, pressure and temperature performs an isothermal flash and returns the volumetric expansivity coefficient (1 / V) * dV / dT in gas phase double v = EStrLVE(integer stream double t, double p) sname xstplve given the stream, pressure and temperature performs an isothermal flash and returns the volumetric expansivity coefficient (1 / V) * dV / dT in liquid phase double hc = EStrHC(integer stream, double t, double p) sname xstphc

given the stream, pressure and temperature performs an isothermal flash and returns the net heat of combustion (gas phase).

double fl = EStrFML(integer stream, double t, double p) sname xstpfml

given the stream, pressure and temperature performs an isothermal flash and returns the flammability lean limit (gas phase).

double fl = EStrFMH(integer stream, double t, double p) sname xstpfmh given the stream, pressure and temperature performs an isothermal flash and returns the flammability rich limit (gas phase).

double s = EStrS(integer stream, double t, double p) sname xstps

given the stream, pressure and temperature performs an isothermal flash and returns the relative entropy (gas + liquid phase)

double d = EStrLD(integer stream, double t, double p)

sname xstpld

given the stream, pressure and temperature performs an isothermal flash and returns the calculated liquid density (at operating conditions).

double d = EStrGD(integer stream, double t, double p) sname xstpgd

given the stream, pressure and temperature performs an isothermal flash and returns the calculated gas density (at operating conditions).

double tc = EStrLC(integer stream, double t, double p)

sname xstplcl

given the stream, pressure and temperature performs an isothermal flash and returns the calculated liquid thermal conductivity (at operating conditions).

double tc = EStrGC(integer stream, double t, double p)

sname xstpgc

given the stream, pressure and temperature performs an isothermal flash and returns the calculated gas thermal conductivity (at operating conditions).

double v = EStrLV(integer stream, double t, double p)

sname xstplv

given the stream, pressure and temperature performs an isothermal flash and returns the r calculated liquid viscosity (at operating conditions).

double v = EStrGV(stream, double t, double p)

sname xstpgv

given the stream, pressure and temperature performs an isothermal flash and returns the calculated gas viscosity (at operating conditions).

double st = EStrST(integer stream, double t, double p)

sname xstpst

given the stream, pressure and temperature performs an isothermal flash and returns the calculated surface tension (at operating conditions).

Methods for chemical's file access

Prode Properties includes a set of functions for accessing data in chemical's file. Components are referenced via a component code which is an integer with value in the range 1 to getFCNR()

Integer nr = getFCNr()sname xfcnr returns the number of components in Chemical's File int str = MCompF(integer code, char *s, integer slm) given the component code fills string s with the relevant component formula (eventually truncated to slm maximum lenght) , this is the Microsoft specific method char *str = CompF(integer code) sname xcf given the component code returns the relevant component formula (eventually truncated to string maximum length), this is the ANSI C compatible method int str = MCompN(integer code, char *s, integer slm) given the component code fills string s with the relevant component name (eventually truncated to slm maximum length), this is the Microsoft specific method char *str = CompN(integer code) sname xcn given the component code returns the relevant component name (eventually truncated to string maximum length), this is the ANSI C compatible method int id = CompID(integer code) sname xcid given the component code returns component's ID (it's the CAS number) int cc = CompCID(integer id) sname xidc given the component ID returns the component's code double mw = CompMw(integer code) sname xcmw given the component code returns the relevant molecular weight double tc = CompTc(integer code) sname xctc given the component code returns the relevant critical temperature double ac = CompAc(integer code) sname xcac given the component code returns the relevant acentric factor double vc = CompVc(integer code) sname xcvc given the component code returns the relevant critical volume double pc = CompPc(integer code) sname xcpc given the component code returns the relevant critical pressure

double dm = CompDm(integer code) sname xcdm given the component code returns the dipole moment double rg = CompRg(integer code) sname xcrg given the component code returns the radius of gyration

double sol = CompSol(integer code) sname xcsol given the component code returns the solubility parameter

double hf = CompHf(integer code) sname xchf given the component code returns the std. enthalpy of formation

double gf = CompGf(integer code) sname xcgf given the component code returns the Gibbs energy of formation

double sf = CompSf(integer code) sname xcsf given the component code returns the enthalpy of fusion

double nb = CompNb(integer code) sname xcnb given the component code returns the normal boiling point

double mp = CompMp(integer code) sname xcmp given the component code returns the melting point

double p = CompVP(integer code, double t)

sname xcvp

given the component code and a temperature, returns the calculated saturation pressure (calculated via Chemical's file temperature dependent correlation)

double h = CompHG(integer code, double t0, double t1) sname xchg

given the component code , initial and final temperatures for integration, returns the calculated ideal gas enthalpy (calculated via Chemical's file temperature dependent correlation)

double s = CompSG(integer code, double t0, double t1)

sname xcsg

given the component code, initial and final temperatures for integration, returns the calculated ideal gas entropy (calculated via Chemical's file temperature dependent correlation)

double h = CompHL(integer code, double t0, double t1)

sname xchl

given the component code , initial and final temperatures for integration, returns the calculated ideal liquid enthalpy (calculated via Chemical's file temperature dependent correlation)

double s = CompSL(integer code, double t0, double t1)

sname xcsl

given the component code , initial and final temperatures for integration, returns the calculated ideal liquid entropy (calculated via Chemical's file temperature dependent correlation)

double h = CompHS(integer code, double t0, double t1) sname xchs

given the component code, initial and final temperatures for integration, returns the calculated ideal solid enthalpy (calculated via Chemical's file temperature dependent correlation)

double s = CompSS(integer code, double t0, double t1) sname xcss given the component code, initial and final temperatures for integration, returns the calculated ideal solid entropy (calculated via Chemical's file temperature dependent correlation) double h = CompHV(integer code, double t) sname xchv given the component code and a temperature, returns the calculated latent heat (calculated via Chemical's file temperature dependent correlation) double v = CompLV(integer code, double t)sname xclv given the component code and a temperature, returns the calculated liquid viscosity (calculated via Chemical's file temperature dependent correlation) double v = CompGV(integer code, double t) sname xcov given the component code and a temperature, returns the calculated gas viscosity (calculated via Chemical's file temperature dependent correlation) double d = CompLD(integer code, double t) sname xcld given the component code and a temperature, returns the calculated liquid density (calculated via Chemical's file temperature dependent correlation) double tc = CompLC(integer code, double t) sname xclc given the component code and a temperature, returns the calculated liquid (thermal) conductivity (calculated via Chemical's file temperature dependent correlation) double tc = CompGC(integer code, double t) sname xcgc given the component code and a temperature, returns the calculated gas (thermal) conductivity (calculated via Chemical's file temperature dependent correlation) double st = CompST (integer code, double t) sname xcst given the component code and a temperature, returns the calculated surface tension (calculated via Chemical's file temperature dependent correlation) double d = CompSD(integer code, double t) sname xcsd given the component code and a temperature, returns the calculated solid density (calculated via Chemical's file temperature dependent correlation) double tc = CompSC(integer code, double t) sname xcsc given the component code and a temperature, returns the calculated solid (thermal) conductivity (calculated via Chemical's file temperature dependent correlation)

Methods to set / access options / settings

To set / access the different options / settings the library includes specific methods,

getKO(), putKO() these methods accept and return a integer (32 bit) which contains all the settings (each bit in the integer represents a different option / setting)

getKS(), putKS() allow to access or define each option

int value = getKO(integer stream) sname xsk given a stream returns a code (integer) with the options

integer res = putKO (integer stream, integer value) sname xsetsk given a stream define the options

int value = getKS(integer stream, integer option) sname xsks given a stream and option (see below the codes) returns a boolean (0-1) with stored value

integer res = putKS (integer stream, integer option, integer value) sname xsetsks given a stream and option define the option.

Table of codes to specify the different options

reference : methods getKO(), setKO() ...

Bit	Decimal value	Option
1	1	set multiphase vapor + liquid
2	2	set multiphase vapor + liquid + solid
3	4	set multiphase vapor + liquid + solid + hydrate
4	8	reduce the number of trial phases (in multiphase)
5	16	use iso compressibility coeff. to detect single phase state
6	32	evaluate stability of each phase in equilibrium
7	64	end specified phase fraction lines when crossing phase boundary lines
8	128	include all hydrate structures (also those not normally generated by formers)

to set one or more options call setOM() passing as value a integer with the sum (decimal values) of all required options.

Table of codes to specify the different states

reference : methods setMP() , PfTF() , PfTF() , StrFv(), StrFvd() ... Code State 0 Vapor phase 1 Liquid phase 2 Solid phase

3 Hydrate phase

Access to specific values

double p = getPatm()
sname xpatm
returns the internal reference (user defined) for atmosferic pressure quantity.

Table of codes to specify the different models

reference : methods setMP(), getMP() ...

Some models may not be available and/or the numerical codes may change in different versions, contact Prode for details

Code	Description	mixing rules	Model
1	Regular		Regular
10	Wilson		Wilson
11	NRTL		NRTL
12	UNIQUAC		UNIQUAC
30	Soave-Redlich_Kwong	VDW	SRK(VDW)
31	Soave-Redlich Kwong ext.	VDW	SRKX(VDW)
40	Soave-Redlich Kwong ext. + NRTL	P-HV	SRKX-NRTL(P-HV)
41	Soave-Redlich Kwong ext. + NRTL	P-LCVM	SRKX-NRTL(P-LCVM)
50	Peng Robinson std.	VDW	PR(VDW)
51	Peng Robinson ext.	VDW	PRX(VDW)
55	Peng Robinson ext. + Wilson	WS	PRX-Wilson(WS)
56	Peng Robinson ext. + UNIQUAC	WS	PRX-UNIQUAC(WS)
57	Peng Robinson ext. + NRTL	WS	PRX-NRTL(WS)
60	Pena Robinson ext. + NRTL	P-HV	PRX-NRTL(P-HV)
61	Peng Robinson ext. + Wilson	P-HV	PRX-Wilson(P-HV)
62	Peng Robinson ext. + UNIQUAC	P-HV	PRX-UNIQUAC(P-HV)
65	Peng Robinson ext. + Wilson	MHV2	PRX-Wilson(MHV2)
66	Peng Robinson ext. + UNIQUAC	MHV2	PRX-UNIQUAC(MHV2)
67	Peng Robinson ext. + NRTI	MHV2	PRX-NRTI (MHV2)
70	Peng Robinson ext + NRTI	P-I CVM	PRX-NRTI (P-I CVM)
71	Peng Robinson ext. + Wilson	P-I CVM	PRX-Wilson(P-I CVM)
72	Peng Robinson ext. + UNIOLIAC	P-I CVM	PRX-UNIOUAC(P-LCVM)
73	Peng Robinson ext. + UNIFAC	P-I CVM	PRX-UNIFAC(WS)
80	Benedict-Webb-Rubin		BWR
81	Benedict-Webb-Rubin-Starling		BWRS
01 00			I K PMI/O
01	Lee Kesler Ploecker		
100			
100	Peng Robinson ext. inc. association (CPA)		
115	Peng Robinson ext. inc. association (CPA) + NRTI		
116	Peng Robinson ext. Inc. association (CPA) + NRTI	P-I CVM	
117	Peng Poblisson ext. inc. association (CPA) + NRTL		PRYCPA NPTI (mod P MHV/2)
110	Peng Robinson ext. inc. association (CPA) + NRTL	WIT V Z	DRYCDA NDTI (D WS)
130		W3	
150	Solid Pure (derived from) PRX-NRTI (P-H\/)		
151	Solid Pure (derived from) PRXCPA-NRTI (P-HV)		SPRXCPA-NRTI (P-HV)
153	Solid Solution (derived from) DRX NRTI (D H)()		
170	Hydrate (derived from) PRYCPA NPTI (P HV)		
170	Hydrate (derived from) PRX NRTI (P HV)		
100			M_{OV}
100	Apphaltana		Achaltana
200	Aspliatene Ditzer (Electrolyte)		
200	Pizer (Liectionyte)		
205	Peng Robinson ext. Ass. / MSA / NRTL		PRACEA-E-INRIL(F-IIV)
200	Stoom tables IADWS 1005		
211			
311 212	GENG 2000 / AGA 2017 ISO 19452 (GEDC)		
31Z			100 10400
315	13U 20103 (AGA 8)		130 20/03

Thermodynamic models

To define or retrieve the thermodynamic models associated with each property (Fg, H, S,V..) of a stream the library includes setMP(), getMP()

integer res = setMP(integer stream, integer mp, integer state, integer model)

sname xsetsm

given a stream, property (Fg,H,S..) model and state (Vapor,Liquid,Solid,Hydrate) this method sets the specified model for that property and returns TRUE in case of success, otherwise returns FALSE

integer m = getMP(integer stream, integer mp, integer state)

sname xsm

given a stream, related property (Fg,H,S..) and state (Vapor,Liquid,Solid,Hydrate) this method returns the specified model for that property and state

Table of codes to specify the different properties in setMP() and getMP()

- 1 Fugacity
- 2 Enthalpy
- 3 Entropy
- 4 Volume
- 5 Viscosity

Base values for enthalpy / entropy

Prode Properties allows the user to define the base values (the temperature and initial value from which to start integration) for entropy and enthalpy from Properties Editor, in setting's page, these values are stored in archive and restored when program starts.

In addition it is possible to modify these value by code with the following methods,

integer res = setHB(integer mod, double t, double val)

sname xsethb

given a code to identify the procedure (see the table with codes), the temperature and initial value sets base value for enthalpy .

integer res = setSB(integer mod, double t, double val)

sname xsetsb

given a code to identify the procedure (see the table with codes), the temperature and initial value sets base value for entropy .

Table of codes to specify the different base values in setHB() and setSB()

- 1 initial values specified by user (values of t and val)
- 2 initial values are enthalpy of formation (or entropy of formation) and temperature 25 C

Stream names

In Prode Properties streams have several properties including a label (name) which could match (for example) the name of a line in your project, you can easily set / access these labels through a series of methods.

integer str = MStrN(integer stream, char *s, integer slm)

given a integer (that identifies a stream) method fills string s with the name of stream (eventually truncated to slm maximum lenght), this is the Microsoft specific method

char *str = StrN(integer stream)

sname xsn

given a integer (that identifies a stream) method returns as ANSI C type the string identifying that stream.

integer res = putN(integer stream, char *str) sname xsetsn

given a integer (that identifies a stream) and a ANSI C string identifying that stream this method sets the label.

Methods to access Model's data

Prode Properties includes models for calculating properties as fugacities, enthalpies, entropies, volumes, viscosities etc. these methods allow to access the models available

integer nr = getMDnr() sname xmdnr returns the number of models available in the library

char *str = getMDN(int model) sname xmdn given the model position (in the range 1-number of models available) the method returns the name of model.

integer res = getMDP(int model, int prop, int state) sname xmdp given the model position (in the range 1-number of models available) the property and state returns TRUE if model can calculate the specified property, otherwise returns FALSE

integer code = getMDC(int model) sname xmdc given the model position (in the range 1-number of models available) returns the code of the model

Methods to control error's messages

The library includes functions to control the error messages

setErrFlag (integer state) sname xseterr given a Boolean (state) se

given a Boolean (state) sets the error flag to TRUE or FALSE. The flag should be cleared (state = FALSE) before each sequence of calculations and tested (method getErrFlag()) after the calcs. If this is done, then a flag state of TRUE indicates that an error has occurred somewhere in the calculation sequence).

integer res = getErrFlag ()

sname xerr

a value of TRUE means that an error has been found, please note that PROPERTIES doesn't clear the error flag state, You should clear the error flag (via setErrFlag()) before each sequence of calc's.

integer str = MErrMsg(char *s, integer slm)

fills string s with the last error message generated (eventually truncated to slm maximum lenght), this is the Microsoft specific method

char *str = ErrMsg() sname xerrmsg Returns the last error message generated, this is the ANSI C compatible method

Methods for accessing Prode Editor

Prode Properties includes methods to open programmatically Properties Editor

integer res = edS(integer stream) sname xeds given a integer (that identifies a stream) method activates the Properties Editor on the specified stream

integer res = edSS() sname xedss this method activates the Properties Editor on first stream

Methods to load / save archives

Archives are files which contain a copy of all stream's, units of measurement, settings etc. stored in Prode Properties memory when the file was created.

When you load an archive all data will be restored, archives are useful to create copies of your work which would otherwise be lost when leaving the application, Prode Properties includes methods for operations on archives.

integer res = AOpen()
sname xaopen
open a file as archive (browse for file)

integer res = AFOpen(char *path) sname xafopen open the file specified in *path as archive

integer res = ASave() sname xasave save a file as archive (browse for file)

integer res = AFSave(char *path) sname xafsave save the file specified in *path as archive

Methods for accessing / defining the units of measurement

Prode Properties includes methods for accessing and defining the units of measurement, these methods utilize a numeric code for identifying the correspondent quantities, refer to the paragraph "Access via software to the units of measurement" for a list of these codes.

integer res = getUMC(integer UM) sname xumc given a integer (that identifies a quantity) method returns the selected UM for that quantity.

integer res = setUMC(integer UM, integer sel) sname xsetumc given two integers (the first identifies a quantity and the second the selection) method selects a UM for that quantity.

integer res = getUMN(integer UM)

sname xumn

given a integer (that identifies a quantity) method returns the number of different units of measurement available for that quantity.

integer str = MgetUMS(integer UM, integer sel, char *s, integer slm) given two integers (the first identifies a quantity and the second the selection) fills string s with selected UM (eventually truncated to slm maximum lenght), this is the Microsoft specific method

char *str = getUMS(integer UM, integer sel)

sname xums

given two integers (the first identifies a quantity and the second the selection) method returns as ANSI C type the string identifying the selected UM.

integer str = MgetSUMS(integer UM, char *s, integer slm) given a integer UM for quantity fills string s with selected UM (eventually truncated to slm maximum lenght), this is the Microsoft specific method

char *str = getSUMS(integer UM) sname xsums given a integer UM for quantity this method returns as ANSI C type the string identifying the selected UM.

double res = UMCR(double value, integer UM, integer SEL) sname xumcr given a value, the code for quantity and selection converts to reference and returns the result

double res = UMCS(double value, integer UM, integer SEL) sname xumcs given a value, the code for quantity and selection converts from reference and returns the result

integer res = UMAU(double a, double b, char *name, integer UM)

sname xumau

given the code for a quantity, the parameters a, b required for conversion and the name adds a new (user defined, temporary) unit.

integer res = UMRAU(integer UM) sname xumrau given the code for a quantity removes all additional (temporary) units

Units of measurement

Prode Properties allows to define via software the units of measurement, see paragraph "Methods for accessing / defining the units of measurement", in Prode Properties to reference a unit must use a numeric code

QUANTITY	UNIT DEF	NUMERIC CODE	DEFAULT UNIT
Pressure (abs)	CONV_P	15	"Pa.a"
Pressure (rel)	CONV_DP	16	"Pa"
Temperature (abs)	CONV_T	17	"K"
Temperature(rel)	CONV_DT	18	"K"
Calorific Value (weight)	CONV_HM	19	"Kj/Kg"
Calorific Value (molar)	CONV_HMM	20	"kj/kmol"
Enthalpy * flow (Streams)	CONV_HS	21	"kW"
Entropy * flow (Streams)	CONV_SS	22	"kJ/(K*s)"
Heat Capacity (weight)	CONV_CP	23	"kJ/(kg*K)"
Heat Capacity (molar)	CONV_CPM	24	"kJ/(kmol*K)"
Flow (mass)	CONV_W	25	"kg/s"
Flow (molar)	CONV_WM	26	"kmol/s"
Flow (volume)	CONV_VW	27	"m3/s"
Density (weight)	CONV_D	28	"kg/m3"
Density (molar)	CONV_DM	29	"kmol/m3"
Specific Volume (weight)	CONV_SV	30	"m3/kg"
Specific Volume (molar)	CONV_SVM	31	"m3/kmol"
Thermal Conductivity	CONV_TC	32	"W/(m*K)"
Viscosity (dynamic)	CONV_V	33	"Pa*s"
Viscosity (kinematic)	CONV_VK	34	"m2/s"
Surface Tension	CONV_ST	35	"N/m"
Lenght	CONV_L	36	"m"
Area	CONV_A	37	"m2"
Volume	CONV_VOL	38	"m3"
Mass	CONV_M	39	"kg"
Velocity	CONV_VL	40	"m/s"
Acceleration	CONV_ACC	41	"m/s2"
Force	CONV_FOR	42	"N"
Time	CONV_TM	43	"s"
Heat Flux	CONV_HF	44	"kW/m2"
Thermal Resistance	CONV_TR	45	"K*m2/kW"
Heat Transfer Coefficent	CONV_HTC	46	"KW/(m2*C)"
Energy	CONV_EN	47	"kJ"
Power	CONV_POW	48	"W"
Dipole moment	CONV_EDM	49	"c-m"
Solubility parameter	CONV_SP	50	"(J/m3)^1/2"
Flow Coefficient	CONV_CV	51	"Cv"
Compressibility coefficient	CONV_CC	52	"1/Pa"
Joule Thomson coefficient	CONV_JTC	53	"K/Pa"
Volume expansivity	CONV_VE	54	"1/K"
Error messages

Prode Properties may generate one or more error messages, herebelow a short list with possible causes

Memory allocation error

· limit in resources allocation (close applications, release memory and restart)

Corrupted file, error reading data file

• the library cannot access a file, this may depend from the file not being in the proper directory or being corrupted, reinstall the software

Internal error

• this error may depend from several different causes, for example a wrong parameter in a function (i.e. an attempt to pass a value out of permitted range).

too many local variables

- too many variables
- a limit in resources allocation

calc. on undefined stream data

• an undefined stream found while executing calc's (edit and define the stream)

undefined stream's operating conditions

• pressure, temperature or flow are undefined (edit and define the stream)

error calling thermo calc. procedure

• wrong input value (calcs cannot converge) or calcs outside temperature range (check chemical's file for limits in temperature correlation's).

cannot converge calc' s loop

• a wrong convergence condition has been specified, for example a parameter outside the correct range etc.

T, P values outside H, S range calcs

• a wrong condition has been specified, for example t, p outside range limits

too many comp's in a stream

- when two or more streams are mixed the total nr. of components may exceed the maximum
- · inconsistencies in stream's data

error accessing component's data archive

• unavailable data (a unspecified component) or calc's outside temperature range.

Stack error (no memory), reload procedure

• a limit in resources allocation (see above)

Method not available in this version

Attempt to define a method not available in that version, edit the stream and define a new method

A stream with Steam Tables model must have only 1 component

· steam tables model requires one component only (water)

Calculation basis

The user can specify which method to use selecting the models. Please refer to the paragraph "reference literature" and "Models" for additional information about the methods.

Fugacity	calculated according selected model
Enthalpy	calculated according selected model
Entropy	calculated according selected model
Volume	calculated according selected model

Viscosity

gas low pressure mixing rule according Wilke (1950), operating conditions correction according Stiel and Thodos (1964) liquid logarithmic average mixing rule, pressure correction according Lucas (1981)

Thermal conductivity

gas low pressure mixing rule according Mason and Saxena (1958), operating conditions correction according Stiel and Thodos (1964) liquid mixing rule according Li (1976)

Surface tension mixing rule according MacLeod-Sugden

Heat of combustion weight average mixing rule according ISO std. (database contains values in Kj/Kg)

Flammability limits mixing rule according Le Chatelier as discussed by Coward & Jones (1952)

Enthalpy, Entropy calc's

the user can specify different initial conditions for enthalpy and entropy, see "Prode Editor : Config Page" for additional details.

Temperature, pressure rangesTemperature range1 K - 5000 KPressure range1 Pa - 1000 Bar

Chemical's File format

This section discusses the file format adopted by Prode Properties to store chemical's data, the program stores for each component a large number of data as shown in following list, data is stored in a binary compressed format.

Prode Properties allows to select different correlations to define each temperature dependent property, all major standards including DIPPR are supported

Note that data dependent correlation's have a range of temperature for application, Prode Properties includes tests for this range (as defined by high and low limits in chemicals file) and, when required, attempts to estimate the values outside this range, in some cases this may produce inconsistent results.

Prode Properties base version adopts the following format

Formula string 12 chars max Name (1) (main list) string 40 chars max Name (2) (user defined list) string 40 chars max Name (3) (user defined list) string 40 chars max Identification number (CAS as default) Molecular weight Critical temperature Critical pressure Critical volume Acentric factor **Dipole Moment** Radius of Gyration Solubility parameter Standard enthalpy of formation (298 K) Gibbs free energy of formation (298 K, 1 atm) Enthalpy of fusion Normal boiling point Melting point Flammability lean limit % (range 0-100) Flammability rich limit % (range 0-100) Autoignition temperature Net heat of combustion Flash Point

Gas heat capacity correlation type of equation unit for property unit for temperature low temperature limit high temperature limit A-E (5 parameters)

Vapor viscosity correlation type of equation unit for property unit for temperature low temperature limit high temperature limit A-E (5 parameters)

Vapor thermal conductivity correlation type of equation unit for property unit for temperature low temperature limit high temperature limit A-E (5 parameters) Heat of vaporization correlation type of equation unit for property unit for temperature low temperature limit high temperature limit A-E (5 parameters)

Liquid vapor pressure correlation type of equation unit for property unit for temperature low temperature limit high temperature limit A-E (5 parameters)

Surface tension type of equation unit for property unit for temperature low temperature limit high temperature limit A-E (5 parameters)

Liquid density correlation type of equation unit for property unit for temperature low temperature limit high temperature limit A-E (5 parameters)

Liquid viscosity correlation type of equation unit for property unit for temperature low temperature limit high temperature limit A-E (5 parameters)

Liquid thermal conductivity correlation type of equation unit for property unit for temperature low temperature limit high temperature limit A-E (5 parameters)

Liquid heat capacity correlation type of equation unit for property unit for temperature low temperature limit high temperature limit A-E (5 parameters) Solid vapor pressure correlation type of equation unit for property unit for temperature low temperature limit high temperature limit A-E (5 parameters)

Solid density correlation type of equation unit for property unit for temperature low temperature limit high temperature limit A-E (5 parameters)

Solid thermal conductivity correlation type of equation unit for property unit for temperature low temperature limit high temperature limit A-E (5 parameters)

Solid heat capacity correlation type of equation unit for property unit for temperature low temperature limit igh temperature limit A-E (5 parameters)

Sources of data

Data in chemical data file come from several sources including :

- "Dechema Chemistry Data ser." text books
- "DIPPR data collection" text books
- "Technical Data Book, Petroleum Refining"

Due to the large differences in critical and transport properties found in different sources, DIPPR (AICHE Design Institute for Physical PRoperty Data) reference has been selected as a default.

Component's identification

Components are identified by name (from DIPPR list), chemical formula and Identification number.

Regression procedures and results

Coefficients in correlations have been calculated with a custom program that uses a modified version of Levenberg-Marquardt algorithm, reported errors (at each fitting point) are usually lower than 1 % of input values for the most complex correlations (i.e. vapor pressure),), however in some cases they may be higher.

Consistency tests

When relations exist between thermodynamic properties (i.e. acentric factor and critical pressure and temperature, vapor pressure and heat of vaporization etc.) a consistency test has been performed.

Comparing Prode Properties results against those of different process simulators

- When comparing data from different tools one must verify that
- the different tools do use the same thermodynamic models
- properties in databanks have siimilar values
- lists and values of BIPs and other parametres which can influence results have similar values

Models

Prode Properties includes a complete set of thermodynamic models (some available in extended versions)

Liquid activities Wilson NTRL UNIQUAC Predictive UNIFAC Electrolytes Pitzer CPA-electrolyte SAFT-electrolyte

Cubic EOS

Soave-Redlich-Kwong, Peng-Robinson with std. alpha function and VdW mixing rules, Extended versions of SRK and PR including parameters calculated to fit experimental data (saturation pressures, densities, heat capacities etc.) and different mixing rules to combine equations of state with activity models Std. and Modified versions of Huron Vidal (HV) rule Std. and Modified versions of Linear Combination of Vidal and Michelsen (LCVM) rule Std. and Modified versions of Michelsen-Huron-Vidal (MHV2) rule Std. and Modified versions of Wong Sandler (WS) rule Other models Modified Benedict-Webb-Rubin Benedict-Webb-Rubin-Starling Lee-Kesler Lee-Kesler-Plocker

Models based on associating fluid theory

Different versions of CPA Cubic Plus Association based on Soave Redlich Kwong and Peng Robinson models with VdW mixing rules and several others to combine equations of state with activity models Std. and Modified versions of Huron Vidal (HV) rule Std. and Modified versions of Linear Combination of Vidal and Michelsen (LCVM) rule etc...

Different versions of SAFT (Perrturbed Chain Statistical Associating Fluid Theory)

Solids

SPM (Solid Pure Model) solid phase treated as single component
SSM (Solid Solution Model) solid phase treated as homogeneous solution
WAX solid phase treated as homogeneous solution (with specific parameters)
Asphaltene
Hydrates (based on Van der Waals and Plateeuw theory with a std. model and a complex model)

Standards (based on international standards) GERG 2008 (ISO 20765) AGA 2017 (2017 version with GERG 2008 formulations) Steam tables (IAPWS 1995) Water / steam properties calculated according IAPWS 1995 formulation

All the models included in Prode Properties export derivatives of Fg, H, S, V vs. W, P, T

UNIFAC functional groups

The underlying idea in UNIFAC method is that a molecule can be considered as a collection of functional groups. The main advantage of this approach is that from a relatively small number of functional groups the properties of many different molecules can be predicted. The UNIFAC model is useful for estimating solution behavior in the absence of experimental data.

Prode Properties incorporates the UNIFAC Group Contribution revision 5 (January 1992, J.P.Baker), following the main groups and subgroups table :

Code	Main	Subgroup	Example
1	CH2	CH3	Hexane
2		CH2	n-Hexane
3		СН	2-Methylpropane
4		С	Neopentane
5	C=C	CH2=CH	1-Hexene
6		CH=CH	2-Hexene
7		CH2=C	2-Methvl-1-butene
8		CH=C	2-Methyl-2-butene
70		C=C	2.3-Dimethylbutene
9	АСН	ACH	Naphthaline
10		AC	Styrene
11	ACCH2	ACCH3	Toluene
12	7100112	ACCH2	FthylBenzene
13		ACCH	Cumene
14	ОН		n-Propanol
15	СНЗОН	CH3OH	Methanol
16			Water
17			Phenol
10			Butanono
10	01200	CH3CO	Pontanono 2
19			Preniania aldahyda
20			Propionic aldenyde
21	0000	CH3COO	Dulyi acetale
22	11000		
23	HCOO	HCOO	Ethyl formate
24	CH2O	CH3O	Dimetnyl ether
25		CH2O	Diethyl ether
26		CHO	Disopropyl ether
27		I HF	letrahydrofuran
28	CNH2	CH3NH2	Methylamine
29		CH2NH2	Ethyl amine
30		CHNH2	Isopropylamine
31	CNH	CH3NH	Dimethylamine
32		CH2NH	Diethyl amine
33		CHNH	Diisopropylamine
34	(C)3N	CH3N	Trimethylamine
35		CH2N	Triethylamine
36	ACNH2	ACNH2	Aniline
37	Pyridine	C5H5N	Pyridine
38		C5H4N	2-Methyl pyridine
39		C5H3N	2,3-Dimethylpyridine
40	CCN	CH3CN	Acetonitrile
41		CH2CN	Propionitrile
42	COOH	COOH	Acetic acid
43		HCOOH	Formic acid
44	CCI	CH2CI	Butane-1-chloro
45		CHCI	Propane-2-chloro
46		CCI	2-Methylpropane-2-chloro
47	CCI2	CH2Cl2	Methane-dichloro
48		CHCl2	Ethane-1,1-dichloro
49		CCI2	Propane-2,2-dichloro
50	CCI3	CHCI3	Chloroform

51		CCI3	Ethane-1,1,1-trichloro
52	CCI4	CCI4	Methane-tetrachloro
53	ACCI	ACCI	Benzene-chloro
54	CNO2	CH3NO2	NitroMethane
55		CH2NO2	Propane-1-nitro
56		CHNO2	Propane-2-nitro
57	ACNO2	ACNO2	Benzene-nitro
58	CS2	CS2	Carbon Disulfide
59	CH3SH	CH3SH	Methanethiol
60		СНОСН	Ethanethiol
61	Furfural	Furfural	Furfural
62			1.2 Ethanedial
62			
64	l Dr	l Pr	Promosthano
04 65			
00	0-0		Hexyne-1
00	DMOO		Hexyne-2
67	DMSO	DMSO	Dimethylsulfoxide
68	ACRY	Acryinitrii	Acryinitrile
69	CICC	CI-(C=C)	Ethene-trichloro
/1	ACF	ACF	Hexafluorobenzene
72	DMF	DMF-1	N,N-Dimethylformamide
73		DMF-2	N,N-Diethylformamide
74	CF2	CF3	Perfluorohexane
75		CF2	
76		CF	Perfluoromethylcyclohexane
77	COO	CO0	Methyl acrylate
78	SiH2	SiH3	Methylsilane
79		SiH2	Diethylsilane
80		SiH	Heptamethyltrisiloxane
81		Si	Heptamethyldisiloxane
82	SiO	SiH2O	1,3-Dimethyldisiloxane
83		SiHO	1,1,3,3-Tetramethyldisiloxane
Code	Main	Subgroup	Example
84		SiO	Octamethylcyclotetrasiloxane
85	NMP	NMP	N-methylpyrrolidone
86	CCIF	CCI3F	Trichlorofluoromethane
87		CCI2F	Tetrachloro-1,2-difluoroethane
88		HCCI2F	Dichlorofluoromethane
89		HCCIF	1-Chloro-1,2,2,2,-tetrafluoroethane
90		CCIF2	1,2-Dichlorotetrafluoroethane
91		HCCIF2	Chlorodifluoromethane
92		CCIF3	Chlorotrifluoromethane
93		CCI2F2	Dichlorodifluoromethane
94	CON	CONH2	Acetamid
95		CONHCH3	N-Methylacetamid
96		CONHCH2	N-Ethylacetamid
97		CON(CH3)2	N,N-Dimethylacetamid
98		CONCH3CH2	N,N-methylethylacetamid
99		CON(CH2)2	N,N-Diethylacetamid
100	OCCOH	C2H5O2	2-Ethoxyethanol
101		C2H402	2-Ethoxy-1-propanol
102	CH2S	CH3S	Dimethylsulfide
103		CH2S	Diethylsulfide
104		CHS	Diisopropylsulfide
105	Morpholine	MORPH	Morpholine
106	Thiophene	C4H4S	Thiophene
107		C4H3S	2-Methylthiophene
108		C4H2S	2,3-Dimethylthiophene
			- 1

Prode Properties, last page